

|  |  |
| --- | --- |
| **oneM2M**  **Technical Specification** | |
| Document Number | oneM2M- TS-0023-V5.3.0 |
| Document Name: | SDT based Information Model and Mapping for Vertical Industries |
| Date: | 2022-09-28 |
| Abstract: | This technical specification includes oneM2M defined information model for home appliances and the mapping with other information models from external organization. |
| Template Version: January 2017 (Do not modify) | |

The present document is provided for future development work within oneM2M only. The Partners accept no liability for any use of this specification.

The present document has not been subject to any approval process by the oneM2M Partners Type 1. Published oneM2M specifications and reports for implementation should be obtained via the oneM2M Partners' Publications Offices.

About oneM2M

The purpose and goal of oneM2M is to develop technical specifications which address the need for a common M2M Service Layer that can be readily embedded within various hardware and software, and relied upon to connect the myriad of devices in the field with M2M application servers worldwide.

More information about oneM2M may be found at: http//www.oneM2M.org

Copyright Notification

© 2018, oneM2M Partners Type 1 (ARIB, ATIS, CCSA, ETSI, TIA, TSDSI, TTA, TTC).

All rights reserved.

The copyright extends to reproduction in all media.

Notice of Disclaimer & Limitation of Liability

The information provided in this document is directed solely to professionals who have the appropriate degree of experience to understand and interpret its contents in accordance with generally accepted engineering or other professional standards and applicable regulations. No recommendation as to products or vendors is made or should be implied.

NO REPRESENTATION OR WARRANTY IS MADE THAT THE INFORMATION IS TECHNICALLY ACCURATE OR SUFFICIENT OR CONFORMS TO ANY STATUTE, GOVERNMENTAL RULE OR REGULATION, AND FURTHER, NO REPRESENTATION OR WARRANTY IS MADE OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR AGAINST INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS. NO oneM2M PARTNER TYPE 1 SHALL BE LIABLE, BEYOND THE AMOUNT OF ANY SUM RECEIVED IN PAYMENT BY THAT PARTNER FOR THIS DOCUMENT, WITH RESPECT TO ANY CLAIM, AND IN NO EVENT SHALL oneM2M BE LIABLE FOR LOST PROFITS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES. oneM2M EXPRESSLY ADVISES ANY AND ALL USE OF OR RELIANCE UPON THIS INFORMATION PROVIDED IN THIS DOCUMENT IS AT THE RISK OF THE USER.

Contents

Contents 3

1 Scope 10

2 References 10

2.1 Normative references 10

2.2 Informative references 11

3 Definitions and Abbreviations 12

3.1 Definitions 12

3.2 Abbreviations 12

4 Conventions 12

5 Harmonised Information Model 12

5.1 Introduction 12

5.2 Design Principle of the Harmonised Information Model 13

5.2.1 Basic design principle of information modelling 13

5.2.2 Description rules for Module Classes and DeviceClasses 13

5.3 ModuleClasses 18

5.3.1 Common Domain 18

5.3.1.1 threeDprinter 18

5.3.1.2 acousticSensor 18

5.3.1.3 airConJobMode 18

5.3.1.4 airFlow 19

5.3.1.5 airPurifierJobMode 19

5.3.1.6 airQualitySensor 19

5.3.1.7 alarmSpeaker 20

5.3.1.8 audioVolume 20

5.3.1.9 autoDocumentFeeder 21

5.3.1.10 battery 21

5.3.1.11 binaryObject 21

5.3.1.12 binarySwitch 22

5.3.1.13 bioElectricalImpedanceAnalysis 22

5.3.1.14 bodyCompositionAnalyser 22

5.3.1.15 boiler 23

5.3.1.16 brewing 23

5.3.1.17 brightness 23

5.3.1.18 clock 24

5.3.1.19 clothesDryerJobMode 24

5.3.1.20 clothesWasherJobMode 24

5.3.1.21 clothesWasherDryerJobMode 24

5.3.1.22 clothesWasherJobModeOption 25

5.3.1.23 colour 25

5.3.1.24 colourSaturation 25

5.3.1.25 connectivity 25

5.3.1.26 cookerHoodJobMode 26

5.3.1.27 credentials 27

5.3.1.28 dehumidifierJobMode 27

5.3.1.29 dishWasherJobMode 27

5.3.1.30 doorStatus 27

5.3.1.31 electricVehicleConnector 28

5.3.1.32 energyConsumption 28

5.3.1.33 energyGeneration 28

5.3.1.34 faultDetection 28

5.3.1.35 filterInfo 29

5.3.1.36 foaming 29

5.3.1.37 galleryMode 29

5.3.1.38 gasChargingControl 29

5.3.1.39 gasMeterAlarm 29

5.3.1.40 gasMeterReportInfo 30

5.3.1.41 geoLocation 30

5.3.1.42 glucometer 31

5.3.1.43 grinder 31

5.3.1.44 heatingZone 31

5.3.1.45 height 32

5.3.1.46 hotWaterSupply 32

5.3.1.47 impactSensor 32

5.3.1.48 keepWarm 32

5.3.1.49 keypad 33

5.3.1.50 liquidLevel 33

5.3.1.51 liquidRemaining 33

5.3.1.52 lock 33

5.3.1.53 mediaSelect 34

5.3.1.54 motionSensor 34

5.3.1.55 numberValue 34

5.3.1.56 openLevel 35

5.3.1.57 operationMode 36

5.3.1.58 overcurrentSensor 36

5.3.1.59 oximeter 36

5.3.1.60 ozoneMeter 36

5.3.1.61 magneticSensorParameters 37

5.3.1.62 parkingStatus 37

5.3.1.63 periodicalReportConfig 37

5.3.1.64 phoneCall 38

5.3.1.65 playerControl 38

5.3.1.66 powerSave 39

5.3.1.67 printQueue 39

5.3.1.68 pulsemeter 39

5.3.1.69 pushButton 40

5.3.1.70 recorder 40

5.3.1.71 refrigeration 40

5.3.1.72 relativeHumidity 41

5.3.1.73 remoteControlEnable 41

5.3.1.74 robotCleanerJobMode 41

5.3.1.75 runState 41

5.3.1.76 securityMode 42

5.3.1.77 sessionDescription 42

5.3.1.78 signalStrength 42

5.3.1.79 slcAlarm 43

5.3.1.80 slcParameterSetting 43

5.3.1.81 slcReportInfo 44

5.3.1.82 smokeSensor 45

5.3.1.83 sphygmomanometer 45

5.3.1.84 spinLevel 45

5.3.1.85 steamClosetJobMode 46

5.3.1.86 televisionChannel 46

5.3.1.87 temperature 46

5.3.1.88 temperatureAlarm 47

5.3.1.89 textMessage 47

5.3.1.90 timer 48

5.3.1.91 turbo 50

5.3.1.92 uvSensor 50

5.3.1.93 waterFlow 50

5.3.1.94 waterMeterAlarm 50

5.3.1.95 waterMeterReportInfo 51

5.3.1.96 waterMeterSetting 52

5.3.1.97 waterSensor 53

5.3.1.98 waterQualityMonitor 53

5.3.1.99 weight 54

5.3.1.100 anemometer 55

5.3.1.101 barometer 55

5.3.1.102 rainGauge 55

5.3.1.103 infraredSensor 55

5.3.1.104 disposal 56

5.3.1.105 waterFilterType 56

5.3.1.106 touchScreen 56

5.3.1.107 prePaidCardReader 56

5.3.1.108 billDeposit 57

5.3.1.109 billWithdrawal 57

5.3.1.110 coinDeposit 57

5.3.1.111 cashDispenser 58

5.3.1.112 cardScanner 58

5.3.1.113 headingSensor 58

5.3.1.114 signalPanel 59

5.3.1.115 crossingBarrier 59

5.3.1.116 threeDDisplay 59

5.3.1.117 threeDScanner 60

5.3.1.118 blender 60

5.3.1.119 shoesWasherJobMode 60

5.3.2 City Domain 61

5.3.3 Health Domain 61

5.3.4 Home Domain 61

5.3.5 Industry Domain 61

5.3.6 Vehicular Domain 61

5.3.7 Agriculture Domain 61

5.3.7.1 cowActivityMonitor 61

5.3.8 Railway Domain 61

5.3.8.1 baliseTransmissionModule 61

5.3.9 Metadata Domain 62

5.3.9.1 features 62

5.3.9.2 location 62

5.3.9.3 localization 62

5.3.9.4 origin 62

5.4 SubDevice models 63

5.4.1 Common Domain 63

5.4.1.1 subDeviceCuff 63

5.4.1.2 subDevicePowerOutlet 63

5.4.2 City Domain 63

5.4.3 Health Domain 63

5.4.4 Home Domain 63

5.4.5 Industry Domain 63

5.4.6 Vehicular Domain 63

5.5 Device models 63

5.5.1 Common Domain 63

5.5.1.1 device3DPrinter 63

5.5.1.2 deviceAirQualityMonitor 64

5.5.1.3 deviceAudioReceiver 64

5.5.1.4 deviceCamera 64

5.5.1.5 deviceDoor 64

5.5.1.6 deviceDoorLock 65

5.5.1.7 deviceLight 65

5.5.1.8 deviceMultiFunctionPrinter 65

5.5.1.9 devicePrinter 65

5.5.1.10 deviceScanner 65

5.5.1.11 deviceSmartPlug 66

5.5.1.12 deviceSwitch 66

5.5.1.13 deviceThermometer 66

5.5.1.14 deviceThermostat 66

5.5.1.15 deviceWaterValve 67

5.5.1.16 deviceServiceButton 67

5.5.1.17 deviceGenericSensor 67

5.5.1.18 device3DDisplay 68

5.5.1.19 device3DScanner 68

5.5.2 City Domain 68

5.5.2.1 deviceOutdoorLamp 68

5.5.2.2 deviceMagneticParkingMonitor 68

5.5.2.3 deviceSmartElectricMeter 69

5.5.2.4 deviceSmartGasMeter 69

5.5.2.5 deviceSmartWaterMeter 69

5.5.2.6 deviceStreetLightController 70

5.5.2.7 deviceWaterQualityMonitor 70

5.5.2.8 deviceWeatherStation 70

5.5.3 Health Domain 70

5.5.3.1 deviceBloodPressureMonitor 70

5.5.3.2 deviceGlucosemeter 71

5.5.3.3 deviceHeartRateMonitor 71

5.5.3.4 devicePulseOximeter 71

5.5.3.5 deviceWeightScaleAndBodyCompositionAnalyser 71

5.5.4 Home Domain 72

5.5.4.1 deviceAirConditioner 72

5.5.4.2 deviceAirPurifier 72

5.5.4.3 deviceClothesDryer 73

5.5.4.4 deviceClothesWasher 73

5.5.4.5 deviceClothesWasherDryer 73

5.5.4.6 deviceCoffeeMachine 74

5.5.4.7 deviceCookerHood 74

5.5.4.8 deviceCooktop 74

5.5.4.9 deviceDehumidifier 74

5.5.4.10 deviceDigitalGallery 75

5.5.4.11 deviceDishWasher 75

5.5.4.12 deviceFan 75

5.5.4.13 deviceFoodProbe 75

5.5.4.14 deviceFreezer 76

5.5.4.15 deviceHomeCCTV 76

5.5.4.16 deviceHumidifier 76

5.5.4.17 deviceKettle 76

5.5.4.18 deviceMicrogeneration 77

5.5.4.19 deviceOven 77

5.5.4.20 deviceRefrigerator 77

5.5.4.21 deviceRiceCooker 77

5.5.4.22 deviceRobotCleaner 78

5.5.4.23 deviceSecurityPanel 78

5.5.4.24 deviceSetTopBox 78

5.5.4.25 deviceSteamCloset 78

5.5.4.26 deviceStorageBattery 79

5.5.4.27 deviceTelevision 79

5.5.4.28 deviceWaterHeater 79

5.5.4.29 deviceWindowShade 79

5.5.4.30 deviceBottleWarmer 80

5.5.4.31 deviceGarbageDisposal 80

5.5.4.32 deviceWaterPurifier 80

5.5.4.33 deviceAirHeater 80

5.5.4.34 deviceJuicer 81

5.5.4.35 deviceShoesWasher 81

5.5.5 Industry Domain 81

5.5.6 Vehicular Domain 81

5.5.6.1 deviceElectricVehicleCharger 81

5.5.6.2 deviceElectricMotorcycle 81

5.5.7 Agriculture Domain 82

5.5.7.1 deviceCowActivityMonitor 82

5.5.8 Railway Domain 82

5.5.8.1 deviceHandheldPTTTerminal 82

5.5.8.2 deviceTrainborneTerminal 82

5.5.8.3 deviceCardRechargingMachine 83

5.5.8.4 deviceSmartGate 83

5.5.8.5 deviceSmartScreenDoor 84

5.6 Enumeration type definitions 84

5.6.1 hd:enum3DprinterTechnology 84

5.6.2 hd:enumAdfState 84

5.6.3 hd:enumAirConJobMode 85

5.6.4 hd:enumAirPurifierJobMode 85

5.6.5 hd:enumAlertColourCode 85

5.6.6 hd:enumCallState 86

5.6.7 hd:enumClothesDryerJobMode 86

5.6.8 hd:enumClothesWasherDryerJobMode 86

5.6.9 hd:enumClothesWasherJobMode 87

5.6.10 hd:enumCookerHoodJobMode 87

5.6.11 hd:enumDehumidifierJobMode 87

5.6.12 hd:enumDishWasherJobMode 88

5.6.13 hd:enumDisplayOrder 88

5.6.14 hd:enumDisplayOrientation 88

5.6.15 hd:enumDoorState 88

5.6.16 hd:enumFoamStrength 89

5.6.17 hd:enumGeneralLevel 89

5.6.18 hd:enumGeneralSpeed 89

5.6.19 hd:enumGeneralTemperature 89

5.6.20 hd:enumGrainsLevel 90

5.6.21 hd:enumGrindCoarseness 90

5.6.22 hd:enumHorizontalDirection 90

5.6.23 hd:enumJobStates 90

5.6.24 hd:enumLiquidLevel 91

5.6.25 hd:enumMachineState 91

5.6.26 hd:enumOzoneStatus 91

5.6.27 hd:enumPlayerMode 92

5.6.28 hd:enumRobotCleanerJobMode 92

5.6.29 hd:enumSecurityMode 92

5.6.30 hd:enumSpinLevelStrength 92

5.6.31 hd:enumSteamClosetJobMode 93

5.6.32 hd:enumSupportedMediaSources 93

5.6.33 hd:enumTasteStrength 93

5.6.34 hd:enumTone 94

5.6.35 hd:enumUvStatus 94

5.6.36 hd:enumVerticalDirection 94

5.6.37 hd:enumWaterFlowStrength 94

5.6.38 hd:enumBaliseSystemIndicator 95

5.6.39 hd:enumWeight 95

5.6.40 hd:enumTemperatureUnit 95

5.6.41 hd:enumWaterFilterType 95

5.6.42 hd:enumDataModelType 95

5.6.43 hd:enumDmAgentState 96

5.6.44 hd:enumFirmwareState 96

5.6.45 hd:enumPowerState 96

5.6.46 hd:enumRebootType 97

5.6.47 hd:enumSoftwareState 97

5.6.48 hd:enumPackageState 97

5.6.49 hd:enumPackageType 97

5.6.50 hd:enumBatteryMaterial 97

5.6.51 hd:enumBatteryShape 98

5.6.52 hd:enum3DDisplayType 98

5.6.53 hd:enum3DScannerType 99

5.6.54 hd:enumShoesWasherJonMode 99

5.7 Universal and Common Properties for Device models 100

5.8 Device Management 100

5.8.1 Introduction 100

5.8.2 flexNode 100

5.8.3 dmAgent 101

5.8.4 dmDeviceInfo 102

5.8.5 dmDataModelIO 102

5.8.6 dmFirmware 103

5.8.7 dmSoftware 106

5.8.8 dmEventLog 107

5.8.9 dmPackage 108

5.8.10 dmAreaNwkInfo 109

5.8.11 dmAreaNwkDeviceInfo 110

5.8.12 dmCapability 110

5.8.13 dmStorage 110

6 The Principle of Resource Mapping for Home Appliance Information Model 111

6.1 Introduction 111

6.2 The Resource Mapping Rules 111

6.2.1 Introduction 111

6.2.2 Resource mapping for Device model 112

6.2.3 Resource mapping for ModuleClass 112

6.2.4 Resource mapping for Action 113

6.2.5 Resource mapping for Property 113

6.2.6 Resource mapping for DataPoint 113

6.2.7 Resource mapping for SubDevice model 114

6.3 Short names 114

6.3.1 Introduction 114

6.3.2 Resource types 114

6.3.3 Resource attributes for properties and data points 119

6.3.4 Resource attributes for actions arguments 124

6.4 containerDefinition values 125

6.4.1 Introduction 125

6.4.2 Device models 125

6.4.3 ModuleClasses 125

6.4.4 Actions 125

6.4.5 SubDevices 125

6.5 XSD definitions 126

6.5.1 Introduction 126

6.5.2 XSD definitions for Device models 126

6.5.3 XSD definitions for ModuleClass 127

6.5.4 XSD definitions for Action 127

6.5.5 XSD definitions for SubDevices 127

6.5.6 XSD definitions for Enumerated Types 127

7 Mapping with Other Information Models from External Organizations 127

7.1 OMA GotAPI(DWAPI) 128

7.1.1 Introduction 128

7.1.2 Device Models 128

7.1.2.1 device3Dprinter 128

7.1.2.2 deviceBloodPressureMonitor 130

7.1.2.3 deviceGlucosemeter 130

7.1.2.4 devicePulseOximeter 131

7.1.2.5 deviceThermometer 132

7.1.2.6 deviceWeightScaleAndBodyCompositionAnalyser 132

7.1.2.7 deviceHeartRateMonitor 133

7.1.3 Data Types 134

8 Ontology for the Home Appliance Information Model aligned with oneM2M Base Ontology 134

A (informative): Resource Mapping Examples 135

A.1 Introduction 135

A.2 Example for Device model ‘deviceAirConditioner' 136

A.3 Example of ModuleClass 'binarySwitch' 138

A.4 Example of Action 'toggle' 139

B (informative): Introduction of External Organizations’ Data Models 141

B.1 OMA Got API(DWAPI-PCH) 141

B.2 OCF 141

B.2.1 Introduction 141

B.2.2 Device Type Mapping 141

C (informative): Mapping to Content Attribute 143

C.1 Introduction 143

C.2 XML representation of SDT instances. 143

C.2.1 Mapping Rules of XML representation 143

C.2.2 Example of XML representation 144

C.3 JSON representation of SDT instances 145

C.3.1 Mapping Rules of JSON representation 145

C.3.2 Example of JSON representation 145

C.4 How to write into *content* attribute 147

D History 148

# Scope

The present document describes the oneM2M defined information model for home appliances, including the description of how it is mapped with other information models from external organizations. It also explains the ontology for the home domain information model.

# References

References are either specific (identified by date of publication and/or edition number or version number) or non‑specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies.

## Normative references

The following referenced documents are necessary, partially or totally, for the application of the present document. Their use in the context of this TS is specified by the normative statements that are referring back to this clause

[1] oneM2M Smart Device Template.

NOTE: Available at https://git.onem2m.org/MAS/SDT

[2] Java coding rule.

[3] oneM2M TS-0001: "Functional Architecture".

[4] oneM2M TS-0004: “Service Layer Core Protocol Specification”

[5] oneM2M TS-0005: ”Management Enablement (OMA)”.

[6] ISO:80000-1: Quantities and units

NOTE: Available at <http://www.oracle.com/technetwork/java/codeconventions-135099.html>.

[7] Open Mobile AllianceTM: “OMA-ER-Device\_WebAPIs-V1\_0-20160419-C”.

NOTE: Available at http://www.openmobilealliance.org/release/DWAPI/V1\_0-20160419-C/OMA-ERELD-DWAPI\_V1\_0-20160419-C.pdf

[8] Open Mobile AllianceTM: “OMA-TS-Blood\_Pressure\_Monitor\_APIs-V1\_0-20160419-C”.

NOTE: Available at http://www.openmobilealliance.org/release/DWAPI/V1\_0-20160419-C/OMA-TS-Blood\_Pressure\_Monitor\_APIs-V1\_0-20160419-C.pdf

[9] Open Mobile AllianceTM: “OMA-TS-Glucometer\_APIs-V1\_0-20160419-C”.

NOTE: Available at http://www.openmobilealliance.org/release/DWAPI/V1\_0-20160419-C/OMA-TS-Glucometer\_APIs-V1\_0-20160419-C.pdf

[10] Open Mobile AllianceTM: “OMA-TS-Heart\_Rate\_Monitor\_APIs-V1\_0-20160419-C”.

NOTE: Available at http://www.openmobilealliance.org/release/DWAPI/V1\_0-20160419-C/OMA-TS-Heart\_Rate\_Monitor\_APIs-V1\_0-20160419-C.pdf

[11] Open Mobile AllianceTM: “OMA-TS-Pulse\_Oximeter\_APIs-V1\_0-20160419-C”.

NOTE: Available at http://www.openmobilealliance.org/release/DWAPI/V1\_0-20160419-C/OMA-TS-Pulse\_Oximeter\_APIs-V1\_0-20160419-C.pdf

[12] Open Mobile AllianceTM: “OMA-TS-Thermometer\_APIs-V1\_0-20160419-C”.

NOTE: Available at http://www.openmobilealliance.org/release/DWAPI/V1\_0-20160419-C/OMA-TS-Thermometer\_APIs-V1\_0-20160419-C.pdf

[13] Open Mobile AllianceTM: “OMA-TS-Weight\_Scale\_Body\_Composition\_Analyzer\_APIs-V1\_0-20160419-C”.

NOTE: Available at http://www.openmobilealliance.org/release/DWAPI/V1\_0-20160419-C/OMA-TS-Weight\_Scale\_Body\_Composition\_Analyzer\_APIs-V1\_0-20160419-C.pdf

[14] W3C Recommendation: “XML Schema Part 2: Datatypes”, 02 May 2001.

NOTE: Available at <http://www.w3.org/XML/Schema/>.

[15] NIST standard FIPS PUB 180-2

[16] IETF RFC 4566: "SDP: Session Description Protocol".

[17] IANA Time Zone Database

NOTE: Available at <https://www.iana.org/time-zones>

[18] Void

[19] Open Mobile AllianceTM: “OMA-ER-GotAPI-V1\_1-20151215-C”.

[20] NIST SP 330:2019: “Special Publication 330 - The International System of Units (SI) 2019 Edition“

NOTE: Available at <https://www.nist.gov/pml/special-publication-330>

[21] oneM2M TS-0033: "Interworking Framework"

## Informative references

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] oneM2M Drafting Rules.

NOTE: Available at <http://www.onem2m.org/images/files/oneM2M-Drafting-Rules.pdf>.

[i.2] oneM2M TR-0017: "Home Domain Abstract Information Model".

[i.3] Void.

[i.4] IEEE 802.15.4: "IEEE Standard for Local and metropolitan area networks--Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)".

[i.5] oneM2M TS-0012: "Base Ontology".

[i.6] <https://en.wikipedia.org/wiki/Multiple_inheritance>

[i.7] <https://www.me.go.kr/home/web/index.do?menuId=10272&condition.code1=007>

[i.8] OCF DEVICE SPECIFICATION V1.3.0

NOTE: Available at <https://openconnectivity.org/specs/OCF_Device_Specification_v1.3.0.pdf>

[i.9] Ju-Hun Park, Hui Sik Kim, Sang-A Hong, Sun Young Jang, “A Study on the Definition of Terms for Domestic Train Control System”, Korean Society for Railway, 2015, http://railway.or.kr/Papers\_Conference/201502/pdf/KSR2015A114.pdf

[i.10] CTCS-3级列控系统总体技术方案 (Overall technology plan Train Control System), 中国铁道出版社(Chinese Railway Press), 2008, ISBN: 9787113091590

[i.11] oneM2M TR-0035: "Developer guide of device management"

# Definitions and Abbreviations

## Definitions

For the purposes of the present document, the following terms and definitions apply:

**Device Class ID:** URN to identify the Device model definition.

**ModuleClass ID:** URN to identify the ModuleClass model definition.

## Abbreviations

For the purposes of the present document, the following abbreviations apply:

ATC Automatic Train Control

ATP Automatic Train Protection

BTM Balise Transmission Module

CTCS-3 Chinese Train Control System-3

GotAPI Generic Open Terminal Application Programming Interface

DWAPI Device Web Application Programming Interface

DWAPI-3DP Device Web Application Programming Interface for 3D printer

DWAPI-PCH Device Web Application Programming Interface for Personal Connected Healthcare

RBC Radio Block Centre

# Conventions

The key words "Shall", "Shall not", "May", "Need not", "Should", "Should not" in the present document are to be interpreted as described in the oneM2M Drafting Rules [i.1].

# Harmonised Information Model

## Introduction

The present document intends to provide the unified means in the oneM2M system by defining a home appliance information model for the home domain devices such as TV, refrigerator, air conditioner, clothes washer, oven, and robot cleaner. For the reasons of interworking with external technologies and efficiency, the principle of the home appliance information model is designed based on HGI SDT 3.0 [1].

The principle of defining the home appliance information model is introduced in clause 5.2. ModuleClasses which oneM2M systems support are explained in clause 5.3. In the subsequent clause 5.5, Device models are defined.

Editor’s note: this clause has to be updated (remove specific references to Home).

## Design Principle of the Harmonised Information Model

Editor’s note: this clause has to be updated (removed specific references to Home).

### Basic design principle of information modelling

The design principle of the oneM2M abstract information model of home appliance, is to use SDT4.0 originally introduced in oneM2M TR‑0017 [i.2]. Note that those terms starting with a capital letter in this clause are SDT terms and are explained in [1].

Domain is a unique name which acts like a namespace (e.g., "org.oneM2M.home.modules"). It is set by the organization creating the SDT, allowing reference to a package of definitions for the contained ModuleClasses and DeviceClass models.

ModuleClasses specifies a single service (e.g., audioVolume, powerOn/Off) with one or more Actions, Properties, DataPoints and Events. Each service which is described as a ModuleClass can be re-used in many DeviceClasses.

DeviceClass model is a physical, addressable, identifiable appliance, sensor and actuator with one or more ModuleClasses, Properties and SubDevices.

SubDevice is a device which may be embedded in a DeviceClass and/or is addressed via another DeviceClass.

Figure 5.2.1‑1 depicts the basic structure of SDT 4.0. Further details about SDT 4.0 and its elements can be found in [1].

Specifications of new DeviceClass models and ModuleClasses are encouraged to re-use the definitions specified in this document as much as possible. If re-use is not possible and new DeviceClass and/or ModuleClases definitions are necessary, it is strongly advised to closely follow the guidelines and definition style from this document.

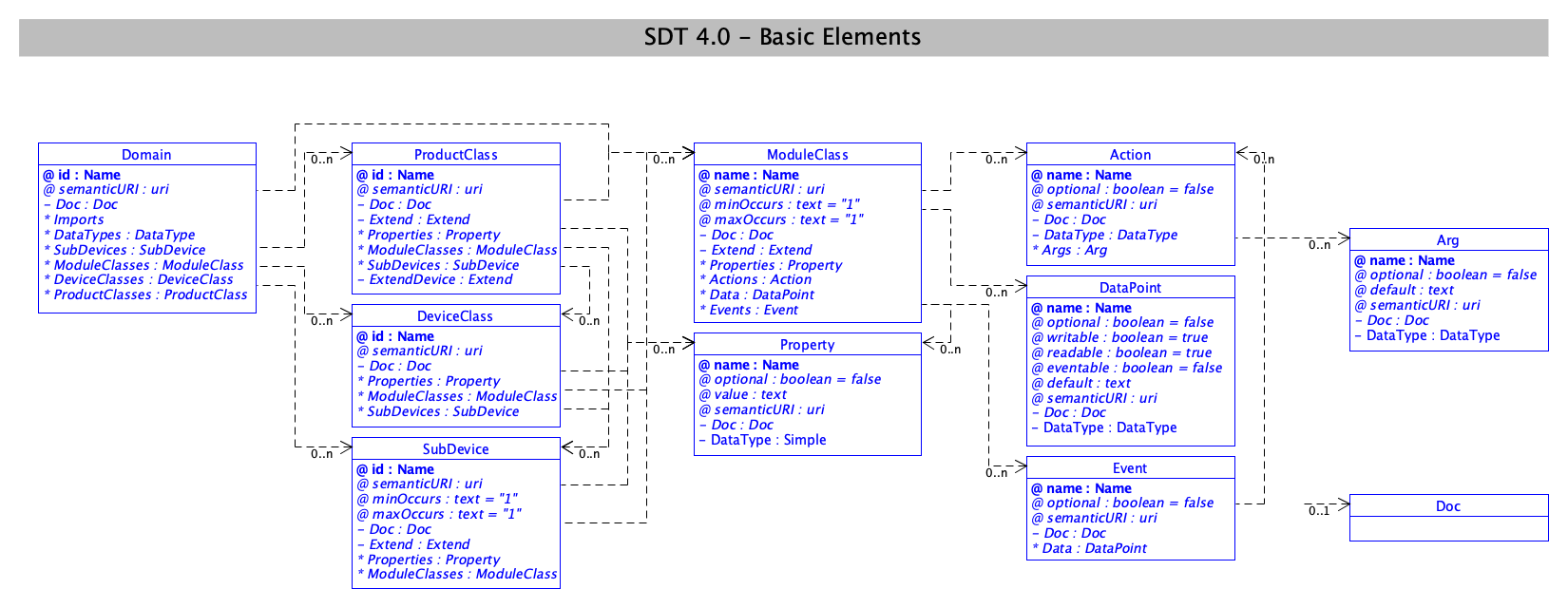


Figure 5.2.1‑1: Design Structure of the Home Appliance Information Model using SDT 4.0

The R/W column of the ModuleClasses’s data point tables in clause 5.3 reflects the intentions of how a data point in a ModuleClass shall be used semantically. This is a “behavioral contract” between applications or users of the modeled devices on the semantic level. Further, the devices or IPE’s (for NoDN) are expected to implement and control the mappings in clause 5.2.2 to implement this “behavioral contract".

### Description rules for Module Classes and DeviceClasses

When the Home Appliances Information Model is described based on SDT, the following rules shall be applied:

* Rule 1: CamelCase rule:
* When naming each element, lowerCamelCase shall be used as the Java coding rules [2].
* Rule 2: Rule for description of Action, DataPoint:
* DataPoint shall be used to represent stateless operations. (e.g. powerState of binarySwitch for on/off operations).
* Action shall be used when describing stateful condition, handling unknown internal state conditions (e.g. upVolume/downVolume by increasing/decreasing the audioVolume in steps, handling transactional procedures, or checking integrity using username plus password at the same time).
* Rule 3: Rule for description of DataPoint and Property:
* Non-functional information shall be described as a Property. Functional information shall be described as a DataPoint. (E.g. non-functional information: version, id; functional information: targetTemperature, targetVolume).
* Rule 4: Definition of the Domain:
* The Domains are specified as “org.onem2m.[domain]”, where [domain] is one of the domain names defined in 6.4.1. The name is chosen according to the domain in which the element is defined.
* The sub-domains for DeviceClasses, SubDevices, ModuleClasses and Actions shall be specified as "org.onem2m.[domain].device", “org.onem2m.[domain].subdevice”, “org.onem2m.[domain].moduleclass”, and “org.onem2m.[domain].action” respectively.
* Rule 5: Naming rule for the element:
* The name of each element should be concise and avoid repeating its parent element name; but
* It may include the name of its parent element for readability. (e.g., lightDimmerUp, lightDimmerDown under lightDimmer).
* All DeviceClasses, SubDevices, ModuleClasses, and Actions of a domain shall be uniquely named.
* Rule 6: Criteria for marking elements as optional or mandatory:
* An element shall only be defined as mandatory if it's foreseen to be universally mandatory to all implementing technologies.
* Rule 7: Enumeration type:
* When describing the meaning of values for enumeration type elements, they may be described under clause 5.6.
* The enumeration types for the harmonized information model are based on <xs:integer>, and the numeric values are interpreted as specified in clause 5.6.
* The name of an enumeration type shall start with the prefix “enum”. This prefix shall not be used with non-enumeration type names.
* All enumeration types are defined under the same domain called Horizontal Domain, which does not contain any other entity. They also must use the same XSD name space identifiers as defined in clause 6.5.1. Even if an enumeration type is used in multiple module classes from different domains, this enumeration type is defined only once.
* Rule 8: Rule for unit in documentation :
* SI (International Systems of Units in [20]) measurement (e.g. meter, kilogram, second.) should be considered as first candidate.
* Otherwise, it may be kept consistency with implementing technologies such as other SDO’s specification.
* Units of measures shall be given in the form of a shortcut compliant to table 5.2.1-1.

Table 5.2.1-1: Shortcuts for units

|  |  |  |
| --- | --- | --- |
| Original name | Short name | Explanation |
| Ampere | A |  |
| Ampere Hour | Ah |  |
| Bar | bar |  |
| Celsius | °C |  |
| Centimeters | cm |  |
| Cubic Meter | m3 |  |
| Cubic meter per hour | m3/h |  |
| Decibel | dB |  |
| Decibel-milliwatts | dBm |  |
| Degrees | deg |  |
| Dots per inch | dpi | dpi is the common unit for spatial dot density |
| g-force | g-f |  |
| Grams | g |  |
| Hertz | Hz |  |
| Kilocalories | kcal |  |
| Kilocalories per hour | kcal/h |  |
| Kilograms per square meter | kg/m2 |  |
| Kilopascal | kPa |  |
| kilovar | kvar |  |
| Kilowatt | kW |  |
| Megabyte | MB | 1 MB = 1024 \* 1024 bytes |
| MegaHertz | MHz |  |
| Meter | m |  |
| Meters per second | m/s |  |
| Miligram per cubic meter | mg/m3 |  |
| Microgram per cubic meter | μg/m3 |  |
| Milligram per deciliter | mg/dl |  |
| Milligram per liter | mg/L |  |
| Millimeter | mm |  |
| Millimeter of mercury | mmHg |  |
| Milliseconds | ms |  |
| Milliwatt per cubic centimetre | mW/cm2 |  |
| Minute | min |  |
| Odor unit per cubic meter | OU/m3 |  |
| Ohm | ohm |  |
| Parts per minute | ppm |  |
| Percent | pct |  |
| Picofarad | pF |  |
| Seconds | s |  |
| Siemens per meter | S/m |  |
| Volt | V |  |
| Watt | W |  |
| Watt hour | Wh |  |



Editor's note: Popular unit in particular industrial domain shall be considered (e.g. cm for human height, calories for energy consumption in healthcare domain). It shall be made coherent in the document, as possible.

* Rule 9: Rule for type :
* Measured and/or calculated values should be represented in float (without taking care of resolution of values).

Editor’s note: It should be made coherent in the document, as possible. Unit shall not be fixed as a rule but be decided with correspondence to each DeviceClass or ModuleClass.

* Rule 10: Inheritance of ModuleClasses :
* A ModuleClass may inherit from another existing ModuleClass in order to provide additional functionalities based on the existing ModuleClass. However, inheritance from multiple ModuleClasses is not allowed (due to the “diamond problem” [i.6]).
* Inheritance of ModuleClass shall only be used in the case that extending an existing ModuleClass is not appropriate, i.e. the functionality to be added is irrelevant to the original design purpose of the existing ModuleClass (e.g. adding a ‘time’ DataPoint to a ‘binarySwitch’ ModuleClass).
* Rule 11: When to differentiate between current and target Data Points in ModuleClasses:
* Device operations, which are executed when setting data points to specific values, may take some time to reach the desired result. For example, setting a new temperature to a heater does not immediately change the room temperature, but it may take some time for the heater to increase the temperature. Therefore, it is sometimes necessary to distinguish between current and target data points.
* A ModuleClass must provide an additional “target” data point when the “current” data point …
  + is writable, and
  + the functionality that is mapped to the data point is an operation, not a configuration function, and
  + the operation may take some time to start and/or to complete, or reach the desired result.
* When a ModuleClass provides current and target data points then the name for the current data point must have the prefix “current”, and the name for the target data point must have the prefix “target”. Both data points must have the same suffix, for example “currentTemperature” and “targetTemperature”.
* Rule 12: Algorithm to generate short names for DeviceClasses, ModuleClasses, Data Points, Actions
* Every domain in oneM2M defines their own short names, i.e. there may exist the same short name in more than one domain, but these short names are distinguished by the domain prefix.
* Previous defined short names of the home domain, e.g. from a previous version of the specification, must be taken into account. They are assigned to the same original names.
* The algorithm to generate the short names from the original names works as follows:
  + The maximum length of a short name for TS-0023 is 5 characters. This length includes the optional appended distinguishing number (see below), but not the suffix for announced resources.
  + If the length of the original name is equal or less than 5 characters, then store the original name as an intermediate result.
  + Else, if the length of the original name is greater than 5 characters, then perform the following procedure:
    - The first and the last character of the original name are stored as first and second character as an intermediate result.
    - All the upper-case characters of the original name, starting with the first upper-case character, are inserted one by one before the last character of the intermediate result, up to a total length of 5 characters of the intermediate result.
    - In case the length of the intermediate result after these steps is less than 5 characters, then the intermediate result is filled with characters from the original string until the length of the intermediate result is 5 characters, following this procedure: the second character of the original name is inserted as the second character of the intermediate result while shifting all characters from the intermediate result by one character forward. This is repeated with the third, fourth, etc., character from the original name.
  + The intermediate result is now compared with all existing short names. If the intermediate result can be found in the list of existing short names, then execute the following steps until the intermediate result cannot be found in the list of previously defined short names:
    - Replace the last character of the intermediate result with an integer number, starting with 0. If the number becomes a two-digit number, then replace the last two characters of the intermediate result, and so forth.
    - Repeat the check described above. If the intermediate result is still the same as an existing short name, then the appended integer number is increased by 1, and the check is repeated.
  + The intermediate result is now stored as a new short name in the list of existing short names.
* Short names for announced resources are created by taking the regular short name of the entity and appending the characters “Annc” to it. Short names for announced resources therefore have a maximum length of 9 characters.

The following table provides some examples for short names that have been created by the described algorithm.

Table 5.2.1-2: Examples for original name to short name mappings

|  |  |
| --- | --- |
| Original name | short name |
| co2 | co2 |
| clock | clock |
| currentJobMode | cuJMe |
| absoluteStartTime | abSTe |
| absoluteStopTime | abST0 |
| impactSensor | impSr |
| impactSensorAnnc | impSrAnnc |

* Rule 13: Rule for R/W column
* The value used in this column defines the interface as it applies to the user of this module. The entity that this module represents (device AE or IPE AE) can read or write to any or all of the datapoints as needed in order to implement the defined interface to the user. <accessControlPolicy> resources shall be defined to enforce access control to the datapoints of the module defined such that R in the R/W column has RETRIEVE accessControlOperations and RW in the R/W column has RETRIEVE and UPDATE accessControlOperations.
* Rule 14: Rule for Optional and Multiplicity
* The value used in the “Optional” column of ModuleClass definitions is mapped to the “optional” element attribute for SDT DataPoint elements.
* The value used in the “Multiplicity” column of DeviceClass and SubDevice definitions is mapped to “minOccurs” and “maxOccurs” element attribute for SDT DeviceClass elements as follows:
  + 1 : minOccurs = 1, maxOccurs = 1
  + 0..1 : minOccurs = 0, maxOccurs = 1
  + 0..N : minOccurs = 0, maxOccurs = unbound
  + 1..N : minOccurs = 1, maxOccurs = unbound

## ModuleClasses

### Common Domain

#### threeDprinter

This ModuleClass provides capabilities for a 3D printer.

Table 5.3.1.1-1: Actions of threeDprinter

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Return Type | Name | Argument | Optional | Documentation |
| none | start3Dprint | none | true | Start 3D printing. |
| none | stop3Dprint | none | true | Stop 3D printing. |

Table 5.3.1.1-2: DataPoints of threeDprinter ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| printType | hd:enum3DprinterTechnology | R | false |  | The type of printing technology (see clause 5.6.1). |
| printSizeX | xs:float | R | false | mm | This data pointrepresents the maximum size of a printing object in the direction of X-axis. |
| printSizeY | xs:float | R | false | mm | This data poin represents the maximum size of printing object in the direction of Y-axis. |
| printSizeZ | xs:float | R | false | mm | This data point represents the maximum size of printing object in the direction of Z-axis. |
| network | xs:boolean | R | false |  | This value indicates the Wide Area Network (WAN) connectivity of the 3D printer, such as Internet or GSM.  ”False” indicates that the printer does not have network connectivity to a WAN. “True” indicates that the printer has WAN network connectivity. |
| memorySize | xs:float | R | false | MB | This value represents the total memory size of the printer. The unit of measure is. |

#### acousticSensor

This ModuleClass provides capabilities for an acoustic sensor.

Table 5.3.1.2-1: DataPoints of acousticSensor ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| loudness | xs:float | R | false | dB | . |
| acousticStatus | xs:integer | R | true | pct | The acousticStatus is expressed in percent, whereas a value of 0 means “no sound” and a value of 100 means “most noisy”. |

#### airConJobMode

This ModuleClasses provides capabilities to control and monitor the job modes of an air conditioner.

Table 5.3.1.3-1: DataPoints of airConJobMode ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| currentJobMode | hd:enumAirConJobMode | RW | false |  | Currently active job mode (see clause 5.6.3). |
| currentJobModeName | xs:string | R | true |  | Name of the current job mode as a string. This can be used when currentJobMode is vendor-specific. |
| jobModes | list of hd:enumAirConJobMode | R | false |  | List of possible job states the device supports (see clause 5.6.3). |

#### airFlow

This ModuleClass provides capabilities for controlling the air flow of a device.

Table 5.3.1.4-1: DataPoints of airFlow ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | Readable | Optional | Unit | Documentation |
| speed | xs:integer | RW | false |  | The current speed level in the range of the [minSpeed, maxSpeed] data poins. |
| minSpeed | xs:integer | R | true |  | The minimum value for the speed level. If not present, the default is 0. |
| maxSpeed | xs:integer | R | true |  | The maximum value for the speed level. If not present, the default is 100. |
| verticalDirection | hd:enumVerticalDirection | RW | true |  | The vertical direction of the air flow (see clause 5.6.36). |
| supportedVerticalDirection | List of hd:enumVerticalDirection | R | true |  | List of supported vertical directions. |
| horizontalDirection | hd:enumHorizontalDirection | RW | true |  | The horizontal direction of the air flow (see clause 5.6.22). |
| supportedHorizontalDirection | List of hd:enumHorizontalDirection | R | true |  | List of supported horizontal directions. |
| automode | xs:Boolean | RW | true |  | Status of the automode feature. “True” indicates that the speed is set by the device, “False” indicates that the device is not controlling the speed. |

#### airPurifierJobMode

This ModuleClasses provides capabilities to control and monitor the job modes of an airPurifier.

Table 5.3.1.5-1: DataPoints of airPurifierJobMode ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| currentJobMode | hd:enumAirPurifierJobMode | RW | false |  | Currently active job mode (see clause 5.6.4). |
| currentJobModeName | xs:string | R | true |  | Name of the current job mode as a string. This can be used when currentJobMode is vendor-specific. |
| jobModes | list of hd:enumAirPurifierJobMode | R | false |  | List of possible job states the device supports (see clause 5.6.4). |

#### airQualitySensor

This ModuleClass provides capabilities for a monitoring sensor that measures the air quality.

Table 5.3.1.6-1: DataPoints of airQualitySense ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| monitoringEnabled | xs:boolean | Rtrue | true |  | The current status of monitoring. "True" indicates enabled, and "False" indicates not enabled. |
| pm1 | xs:integer | R | true | µg/m3 | The concentration of particle matter under 1㎛.. The minimum value is 0. |
| pm25 | xs:integer | R | true | µg/m3 | The concentration of particle matter under 2.5㎛. The minimum value is 0. |
| pm10 | xs:integer | R | true | µg/m3 | The concentration of particle matter under 10㎛. The minimum value is 0. |
| tsp | xs:float | R | true | µg/m3 | Total suspended particle. |
| odor | xs:integer | R | true | OU/m3 | The concentration of odor that reflects air pollution. The minimum value is 0. |
| humidity | xs:float | R | true | pct | The measured humidity. The minimum value is 0, and the maximum value is 100. |
| temperature | xs:float | R | ture | °C | The current temperature |
| airPressure | xs:float | R | ture | KPa | The air pressure. |
| co | xs:float | R | true | mg/m3 | This value indicates the CO level. |
| co2 | xs:float | R | true | mg/m3 | This value indicates the CO2 level. |
| ch2o | xs:float | R | true | µg/m3 | This value indicates the CH2O level. |
| voc | xs:float | R | true | ppm | This value indicates the VOC (Volatile Organic Compounds). |
| no2 | xs:float | R | true | µg/m3 | This value indicates the concentration of NO2. |
| so2 | xs:float | R | true | µg/m3 | This value indicates the concentration of SO2. |
| o3 | xs:float | R | true | µg/m3 | This value indicates the concentration of O3. |
| noise | xs:float | R | true | dB | This value indicates the level of noise. |
| windDirection | xs:float | R | true | deg | The wind direction. The value range is [0-359]. North is 0.0 degrees, east is 90.0 degrees, south is 180.0 degrees, west is 270.0 degrees. |
| windSpeed | xs:float | R | true | m/s | The wind speed |

#### alarmSpeaker

This ModuleClass provides the capabilites to initiate and monitor an alarm.

Table 5.3.1.7-1: DataPoints of alarmSpeaker ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| tone | hd:enumTone | RW | true |  | Representing the tones of the alarm (see clause 5.6.34). |
| light | hd:enumAlertColourCode | R | true |  | Representing the lighting mode of the alarm (see clause 5.6.5). |
| alarmStatus | xs:boolean | R | false |  | "True" indicates the alarm start while "False" indicates the alarm stop. |

#### audioVolume

This ModuleClass provides capabilities to control and monitor volume

Table 5.3.1.8-1: Actions of audioVolume

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Return Type | Name | Argument | Optional | Documentation |
| none | upVolume | none | true | Increase the volume by the amount of the stepValue up to the maxValue. |
| none | downVolume | none | true | Decrease the volume by the amount of the stepValue down to 0. |
| none | muteAudioVolume | none | true | Save the current audio volume and set the volume to 0. |

Table 5.3.1.8-2: DataPoints of audioVolume

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| volumeValue | xs:integer | RW | false | pct | The rounded percentage of the current volume in the range of [0, maxValue]. 0 percentage shall mean no sound produced. |
| stepValue | xs:integer | R | true | pct | Step value used by the “UpVolume” and “DownVolume” actions. |
| maxValue | xs:integer | R | true | pct | Maximum value allowed for Volume. maxValue is 100 by default if “maxValue” is not provided. |
| muteEnabled | xs:boolean | RW | false |  | The current status of the mute enablement. "True" indicates enabled (that is, no sound), and "False" indicates not enabled (that is, sound is played). |
| saveAudioVolume | xs:integer | RW | true | pct | Save the current audio volume for the muteAudioVolume |

#### autoDocumentFeeder

This ModuleClasses provides capabilities to monitor the state of autoDocumentFeeder (ADF). ADF is a feature which takes several pages and feeds the paper one page at a time into a scanner or printer, allowing the user to scan, print or fax, multiple-page documents without having to manually replace each page.

Table 5.3.1.9-1: DataPoints of autoDocumentFeeder ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| currentAdfState | hd:enumAdfState | R | false |  | Current state of the ADF. |
| adfStates | list of hd:enumAdfState | R | false |  | List of possible ADF states that are supported by the device (see clause 5.6.2). |

#### battery

This ModuleClass provides capabilities to indicate the detection of low battery and gives an alarm if the triggering criterion is met. The level data point in the ModuleClass represents the current battery charge level.

Table 5.3.1.10-1: DataPoints of battery ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| level | xs:integer | R | false | pct | The rounded percentage of the current charging level of a battery in the range of [0, 100]. 0 percentage shall mean that no charge remains. |
| capacity | xs:integer | R | true | mAh | The total capacity of battery in mAh. |
| rechargeable | xs:boolean | R | true |  | To indicate the battery is rechargeable or not. “True” inducates rechargeable. |
| charging | xs:boolean | R | True |  | The status of charging. "True" indicates enabled, and "False" indicates not enabled. |
| discharging | xs:boolean | R | True |  | The status of discharging. "True" indicates charging, and "False" indicates not charging. |
| lowBattery | xs:boolean | R | True |  | To indicate that the battery is on a low charge level. |
| batteryThreshold | xs:integer | RW | True |  | When a battery’s “level” is less than “batteryThreshold” then “lowBattery” is set to “True”. This datapoint can be used to raise an alarm, depending on the implementation. |
| chargingVoltage | xs:float | R | true | V | The voltage to charge the battery |
| chargingAmpere | xs:float | R | true | A | The ampere to charge the battery |
| dischargingVoltage | xs:float | R | true | V | The voltage to discharge the battery |
| dischargingAmpere | xs:float | R | true | A | The ampere to discharge the battery |
| batteryMaterial | hd:enumBatteryMaterial | R | true |  | The material of the cell of the battery |
| batteryShape | hd:enumBatteryShape | R | true |  | The size of the battery such as “AAA”. |

#### binaryObject

This ModuleClass describes the handling of a binary object (blob).

Table 5.3.1.11-1: DataPoints of binaryObject ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| object | xs:string | RW | false |  | This data point contains the base64 encoded binary object. |
| objectType | xs:string | RW | false |  | This data point contains the type and subtype of the binary object as a MIME type. |
| size | xs:integer | RW | true |  | The size of the decoded binary object. |
| hash | xs:string | RW | true |  | The hash code of the blob. If present, it is used to check the decoded content of the “object” data point for integrity. The algorithm used for generating the hash value is SHA-2 [15]. The data point contains the hash as a hex encoded value. |

#### binarySwitch

This ModuleClass provides capabilities to control and monitor the state of power.

Table 5.3.1.12-1: Actions of binarySwitch ModuleClass

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Return Type | Name | Argument | Optional | Documentation |
| none | toggle | none | true | Toggle the switch. |

Table 5.3.1.12-2: DataPoints of binarySwitch ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| state | xs:boolean | RW | false |  | The current status of the binarySwitch. "True" indicates turned-on, and "False" indicates turned-off. |

#### bioElectricalImpedanceAnalysis

This ModuleClass provides the analysis of human body tissue based on impedance measurement.

Table 5.3.1.13-1: DataPoints of bioElectricalImpedanceAnalysis ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| water | xs:float | R | false | pct | The water content measurement from the BIA. |
| fat | xs:float | R | false | pct | The fat content measurement from the BIA. |
| muscle | xs:float | R | false | pct | The muscle content measurement from the BIA. |
| bone | xs:float | R | false | pct | The bone content measurement from the BIA. |
| visceraFat | xs:float | R | false | pct | The viscera fat content measurement from the BIA. |
| kcal | xs:float | R | false | kcal | The kcal (kilocalories) measurement from the BIA. |
| resistance | xs:float | R | false | ohm | The resistance of human body. |

#### bodyCompositionAnalyser

This ModuleClass provides the capability to report the measurement of body composition analyser characteristics.

Table 5.3.1.14-1: DataPoints of body composition analyser ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| bodyLength | xs:float | R | true | cm | The measurement of body length by Weight scale and Body composition analyser. |
| bmi | xs:float | R | true | kg/m2 | The measurement of Body Mass Index (BMI) by a weight scale and a body composition analyser. |
| fatFreeMass | xs:float | R | true | kg | The measurement of fat free mass by a weight scale and a body composition analyser. |
| softLeanMass | xs:float | R | true | kg | The measurement of soft lean mass by a weight scale and a body composition analyser. |
| muscleMass | xs:float | R | true | kg | The measurement of muscle mass by a weight scale and a body composition analyser. |
| basalMetabolism | xs:float | R | true | kcal | The measurement of basal metabolism by a weight scale and a body composition analyser. |
| impedance | xs:float | R | true | ohm | The measurement of impedance by a weight scale and a body composition analyser. |
| [protein](file:///D:\Program%20Files\Youdao\Dict\7.5.0.0\resultui\dict\?keyword=protein)Mass | xs:float | R | true | kg | The measurement of [protein](file:///D:\Program%20Files\Youdao\Dict\7.5.0.0\resultui\dict\?keyword=protein) mass by a weight scale and a body composition analyser. |
| bodyWaterMass | xs:float | R | true | kg | The measurement of body water mass by a weight scale and a body composition analyser. |
| inorganicSaltMass | xs:float | R | true | g | The measurement of inorganic salt mass by a weight scale and a body composition analyser. |
| somatotype | xs:string | R | true |  | The measurement of somatotype by Weight scale and Body composition analyser. |

#### boiler

This ModuleClass provides capabilities to control the status of the boiling functionality for water heaters.

Table 5.3.1.15-1: DataPoints of boiler ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R-W | Optional | Unit | Documentation |
| status | xs:boolean | RW | false |  | The status of boiling. “True” indicates boiling, “False” indicates not boiling. |

#### brewing

This ModuleClass provides capabilities to control and monitor a brewing process. It is intended to be part of devices that prepare hot drinks such as a coffee or a tea.

Table 5.3.1.16-1: DataPoints of brewing ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| cupsNumber | xs:integer | RW | false |  | The current number of the cups requested to brew. |
| strength | hd:enumTasteStrength | RW | true |  | The current strength of the drink taste (see clause 5.6.33). A higher value indicates a stronger taste. |

#### brightness

This ModuleClass provides capabilities to control and monitor the brightness of a light for example from a lamp. Brightness is scaled as a percentage. A lamp or a monitor can be adjusted to a level of light between very dim (0% is the minimum brightness) and very bright (100% is the maximum brightness).

Table 5.3.1.17-1: DataPoints of brightness ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| brightness | xs:integer | RW | false | pct | The status of brightness level. |

#### clock

This ModuleClass provides capabilities to control and monitor time and date information.

Table 5.3.1.18-1: DataPoints of clock ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| currentTime | xs:time | RW | false |  | Information of the current time |
| currentDate | xs:date | RW | false |  | Information of the current date |
| currentTimeZone | xs:string | RW | true |  | Name of current time zone according to the IANA Timezone data format (TZ) [17], for example, “[America/New York](https://en.wikipedia.org/w/index.php?title=America/New_York&action=edit&redlink=1)”. |

#### clothesDryerJobMode

This ModuleClasses provides capabilities to control and monitor the job modes of a clothes dryer.

Table 5.3.1.19-1: DataPoints of clothesDryerJobMode ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| currentJobMode | hd:enumClothesDryerJobMode | RW | false |  | Currently active job mode (see clause 5.6.7). |
| currentJobModeName | xs:string | R | true |  | The name of current job mode as a string. This can be used when “currentJobMode” is vendor-specific. |
| jobModes | list of hd:enumClothesDryerJobMode | R | false |  | List of possible job states the device supports. |

#### clothesWasherJobMode

This ModuleClasses provides capabilities to control and monitor the job mode of a washer.

Table 5.3.1.20-1: DataPoints of clothesWasherJobMode ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| currentJobMode | hd: enumClothesWasherJobMode | RW | false |  | Currently active job mode (see clause 5.6.8). |
| currentJobModeName | xs:string | RW | true |  | The name of the current job mode as a string. This can be used when the currentJobMode is vendor-specific. |
| jobModes | list of hd:enumClothesWasherJobMode | R | false |  | List of possible job states that the device supports (see clause 5.6.8). |

#### clothesWasherDryerJobMode

This ModuleClasses provides capabilities to control and monitor the job modes of clothesWasherDryer.

**Table 5.3.1.21-1: DataPoints of clothesWasherDryerJobMode ModuleClass**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **R/W** | **Optional** | **Unit** | **Documentation** |
| currentJobMode | hd: enumClothesWasherDryerJobMode | RW | false |  | Currently active job (see clause 5.6.8). |
| currentJobModeName | xs:string | RW | true |  | The name of the current job mode as a string. This can be used when currentJobMode is vendor-specific. |
| jobModes | list of hd:enumClothesWasherDryerJobMode | R | false |  | List of possible job states the device supports (see clause 5.6.8). |

#### clothesWasherJobModeOption

This ModuleClasses provides capabilities to control and monitor the washing job mode options of a washer.

Table 5.3.1.22-1: DataPoints of clothesWasherJobModeOption ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| washTemp | hd:enumGeneralTemperature | RW | true |  | This data point represents the water temperature level (see clause 5.6.19). |
| soilLevel | hd:enumGeneralLevel | RW | true |  | This data point represents the washing level (see clause 5.6.17). |
| spinSpeed | hd:enumGeneralSpeed | RW | true |  | This data point represents the spin-dry speed level (see clause 5.6.18). |
| preWash | xs:boolean | RW | true |  | This data point indicates pre-wash. “True” indicates enabled, “False” indicates disabled. |
| speedWash | xs:boolean | RW | true |  | This data point indicates speed wash. “True” indicates enabled, “False” indicates disabled. |
| steamTreat | xs:boolean | RW | true |  | This data point indicates steam treat. “True” indicates enabled, “False” indicates disabled. |
| coldWash | xs:boolean | RW | true |  | This data point indicates cold wash. “True” indicates enabled, “False” indicates disabled. |
| extraRinse | xs:boolean | RW | true |  | This data point indicates extra rinse. “True” indicates enabled, “False” indicates disabled. |

#### colour

This ModuleClass provides the capabilities to set the value of the Red, Green, and Blue colour channels for a colour device.

Table 5.3.1.23-1: DataPoints of colour ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| red | xs:integer | RW | false |  | The value of the Red colour channel of RGB. The range is [0,255]. |
| green | xs:integer | RW | false |  | The value of the Green colour channel of RGB. The range is [0,255]. |
| blue | xs:integer | RW | false |  | The value of the Blue colour channel of RGB. The range is [0,255]. |

#### colourSaturation

This ModuleClass provides cababilities to control and monitor a colour saturation value.

Table 5.3.1.24-1: DataPoints of colourSaturation ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| colourSaturation | xs:integer | RW | false | pct | The status of colour saturation level. “colourSaturation” has a range of [0,100]. A “colourSaturation” value of 0 means that a device displays or produces black and white images. A “colourSaturation” value of 50 means that a device displays or produces normal colour images. A “colourSaturation” value of 100 means that a device displays or produces very colourfull images. |

#### connectivity

This ModuleClass provides capabilities to monitor network connectivity.

Table 5.3.1.25-1: DataPoints of connectivity ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| rsrp | xs:integer | R | false | dBm | Reference signal received power (RSRP) is a measurement of the received power level in an [LTE](https://en.wikipedia.org/wiki/LTE_(telecommunication)) cell network. |
| rsrq | xs:integer | R | true | dB | Reference signal received quality (RSRQ) indicates the quality of the received reference signal. RSRQ is defined as the ratio N×RSRP/(E-UTRA carrier RSSI), where N is the number of RB’s of the E-UTRA carrier RSSI measurement bandwidth. The measurements in the numerator and denominator shall be made over the same set of resource blocks. |
| cellID | xs:integer | R | true |  | Serving Cell ID in case Network Bearer Resource is a Cellular Network. |
| rssi | xs:integer | R | true | dBm | In telecommunications, received signal strength indicator (RSSI) is a measurement of the power present in a received radio signal. |
| signalECL | xs:integer | R | true |  | Based on measurements of the reference signal’s received power, the UE will select an entry coverage enhancement level (ECL) to camp into the cell. The coverage level will determine the Narrowband Physical Random Access Channel (NPRACH) resources used by the device and will inform the eNB of the device receiver sensitivity conditions. |
| sinr | xs:integer | R | true | dB | Signal to interference plus noise ratio (SINR) is commonly used in wireless communication as a way to measure the quality of wireless connections. |
| pci | xs:string | R | true |  | Physical Cell ID is an indentification of a cell at physical layer. |
| dailyActivityTime | xs:integer | R | true | s | Daily communication time (Starts at 00:00h) |
| dailyNumberOfConnections | xs:integer | R | true |  | Daily number of connections (Starts at 00:00h) |
| commFreqValue | xs:integer | R | true | MHz | Communication frequency value (commFreqValue) is the transmission frequency of the wireless signal. |
| currentCycleBeginn | xs:datetime | R | true |  | A timestamp that indicates the beginning of the current cycle for counting the transfer volumina and transmission errors. |
| currentCycleVolume | xs:integer | R | true | bytes | Number of bytes transferred since currentCycleBeginn |
| currentCycleTransmissionErrors | xs:integer | R | true |  | Number of transmission errors since currentCycleBeginn |
| minimumCommunicationLatency | xs:integer | R | true | s | The minimum time delay between the last communication attempt |

#### cookerHoodJobMode

This ModuleClasses provides capabilities to control and monitor the job modes of a cookerHood.

Table 5.3.1.26-1: DataPoints of cookerHoodJobMode ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| currentJobMode | hd: enumCookerHoodJobMode | RW | false |  | The currently active job mode. |
| currentJobModeName | xs:string | R | true |  | The name of the current job mode as a string. This can be used when currentJobMode is vendor-specific. |
| jobModes | list of hd: enumCookerHoodJobMode | R | false |  | List of possible job states the device supports (see clause 5.6.10). |

#### credentials

This ModuleClass provides the capability to manage user credentials which allows a user to authenticate on an appliance or a server that is associated with the appliance. The authentication depends on a user login and password, or on a token. An example appliance which may include this ModuleClass is a camera.

Table 5.3.1.27-1: DataPoints of credentials ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| loginName | xs:string | W | true |  | The user’s login name. |
| password | xs:string | W | true |  | The user’s password. |
| token | xs:string | W | true |  | An authentication token, for example an OAuth token. |

#### dehumidifierJobMode

This ModuleClasses provides capabilities to control and monitor the job modes of a dehumidifier device.

Table 5.3.1.28-1: DataPoints of dehumidifierJobMode ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| currentJobMode | hd:enumDehumidifierJobMode | RW | false |  | The currently active job mode (see clause 5.6.11). |
| currentJobModeName | xs:string | R | true |  | The name of the current job mode as a string. This can be used when currentJobMode is vendor-specific. |
| jobModes | list of hd:enumDehumidifierJobMode | R | false |  | List of possible job states the device supports. |

#### dishWasherJobMode

This ModuleClasses provides capabilities to control and monitor the job modes of a dishWasher.

Table 5.3.1.29-1: DataPoints: DataPoints of dishWasherJobMode ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| currentJobMode | hd: enumDishWasherJobMode | RW | false |  | The currently active job mode. |
| currentJobModeName | xs:string | R | true |  | The name of the current job mode as a string. This can be used when the currentJobMode is vendor-specific. |
| jobModes | list of hd: enumDishWasherJobMode | R | false |  | List of possible job states the device supports (see clause 5.6.12) |

#### doorStatus

This ModuleClass provides the status of a door. It is intended to be part of a device such as a refrigerator and an oven that might have multiple doors.

Table 5.3.1.30-1: DataPoints of doorStatus ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| doorState | hd:enumDoorState | R | false |  | Current state of the door (see clause 5.6.15). |
| openDuration | m2m:timestamp | R | true |  | The time duration the door has been open. |
| openAlarm | xs:boolean | RW | true |  | The state of the door open alarm. "True" indicates that the open alarm is active. "False" indicates that the open alarm is not active. |

#### electricVehicleConnector

This ModuleClass provides information about charging/discharging devices for electric vehicles.

Table 5.3.1.31-1: DataPoints of electricVehicleConnector ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| Status | xs:boolean | R | false |  | The status of connection. “True” means connected, “False” means not connected. |
| chargingCapacity | xs:integer | R | true | Ah | Rated charging capacity. |
| dischargingCapacity | xs:integer | R | true | Ah | Rated discharging capacity. |

#### energyConsumption

This ModuleClass describes the measured energy consumed by the device since power up. One particular use case for the energyConsumption ModuleClass is a smart meter.

Table 5.3.1.32-1: DataPoints of energyConsumption ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| power | xs:float | R | true | W | The power of the device. |
| absoluteEnergyConsumption | xs:float | R | false | Wh | The absolute energy consumption, reflecting the real measurement of accumulative energy. |
| roundingEnergyConsumption | xs: integer | R | true |  | This energy consumption data is calculated by multiplying significantDigits with multiplyingFactors, and rounding down the result. |
| significantDigits | xs:integer | R | true |  | The number of effective digits for data. |
| multiplyingFactors | xs:float | R | true |  | The unit for data multiplying factors, for example 1 kWh, 0,1 kWh, 0,01 kWh etc. |
| voltage | xs:float | R | true | V | The voltage of the device. |
| current | xs:float | R | true | A | The current of the device. |
| frequency | xs:float | R | true | Hz | The frequency of the device. |
| measuringScope | xs:string | RW | true |  | The measuring scope of the meter, for example the whole house, a room, or a device. |

#### energyGeneration

This ModuleClass provides information about generation data on electric generator devices such as a photo voltaic power system, fuel cells, or microgeneration.

Table 5.3.1.33-1: DataPoints of energyGenerationModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| powerGenerationData | xs:float | R | true | W | Amount of instaneous generation data. |
| roundingEnergyGeneration | xs:integer | R | true |  | This energy consumption data is calculated by multiplying significantDigits with multiplyingFactors, and rounding down the result. |
| significantDigits | xs:integer | R | true |  | The number of effective digits for data. |
| multiplyingFactors | xs:floatr | R | true |  | The unit for data multiplying factors, for example 1 kWh, 0,1 kWh, 0,01 kWh etc. |
| generationSource | xs:string | RW | false |  | The type of generating source. |

#### faultDetection

This ModuleClass provides information about whether a fault has occurred in a device.

Table 5.3.1.34-1: DataPoints of faultDetection ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| status | xs:boolean | R | false |  | The status of fault detection. |
| code | xs:integer | R | true |  | The numeric representation of the fault. |
| description | xs:string | R | true |  | The message representation of the fault. |

#### filterInfo

This ModuleClass is for monitoring filter information of a device.

Table 5.3.1.35-1: DataPoints of filterInfo ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| usedTime | xs:integer | R | false |  | The cumulative used time in seconds of a filter |
| needsReplacement | xs:boolean | R | true |  | This value indicates that the filter needs to be replaced. |
| filterLifetime | xs:integer | R | true | pct | Percentage life time remaining for the water filter. |

#### foaming

This ModuleClass provides capabilities to control and monitor desired parameters of foam e.g. for foaming milk. It is initially intended to be part of a device that prepare drinks with milk (for example a coffee machine or hot chocolate machine).

Table 5.3.1.36-1: DataPoints of foaming ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| foamingStrength | hd:enumFoamStrength | RW | false |  | The current strength of foamed milk. A higher value indicates more foamed milk (see clause 5.6.16) |

#### galleryMode

This ModuleClass provides information about the mode of display. The galleryMode includes the display orientation, display interval and display order.

Table 5.3.1.37-1: DataPoints of galleryMode ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| displayOrientation | hd:displayOrientation | RW | false |  | The orientation of display |
| displayInterval | xs:integer | RW | true | s | The interval of changing display content. |
| displayOrder | hd:displayOrder | RW | true |  | The sequence of the displaying content |

#### gasChargingControl

This ModuleClass provides capabilities to recharge the gas meter.

Table 5.3.1.38-1: DataPoints of gasChargingControl ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| billingNumber | xs:string | RW | true |  | The billing number of the remote recharge. |
| rechargeCredit | xs:float | RW | true |  | Amount of recharge money in local currency. |
| surplusCredit | xs:float | R | true |  | Amount of surplus money in local currency. |
| rechargeGas | xs:float | RW | true | m3 | Amount of recharge gas. |
| surplusGas | xs:float | R | true | m3 | Amount of surplus gas. |

#### gasMeterAlarm

This ModuleClass provides capabilities to set service parameters of a gas meter.

Table 5.3.1.39-1: DataPoints of gasMeterAlarm ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| lowGasAlarm | xs:boolean | R | true |  | The alarm of insufficient amount of gas. |
| lowCreditAlarm | xs:boolean | R | true |  | The alarm of insufficient amount of money. |
| leakageAlarm | xs:boolean | R | true |  | The alarm of gas leakage. |
| lockedDownAlarm | xs:boolean | R | true |  | The alarm of the gasmeter being locked. When the gas meter is in use, the lower wheel does not work, called the "dead meter". |
| largeFlowAlarm | xs:boolean | R | true |  | The alarm of large flow occurring. The instantaneous flow passed exceeds the setting maximum flow of the gasmeter. |
| magneticDisturb | xs:boolean | R | true |  | The alarm of magnetic interference occurring. |
| singleCountAlarm | xs:boolean | R | true |  | The alarm of single reed switch counting. The normal condition is that the double reed switches are working. If only a single reed switch is working, it means that the metering has a problem and shall fire the alarm. |

#### gasMeterReportInfo

This ModuleClass provides information of measurements of a gas meter.

Table 5.3.1.40-1: DataPoints of gasMeterReportInfo ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| totalUseValue | xs:float | R | false | m3 | The total consumption of gas. |
| valveStatus | xs:boolean | R | true |  | The status of gasmeter valve.   * “True”: open * “False”: close |

#### geoLocation

This ModuleClass provides the capability to get or set geo-location information.

Table 5.3.1.41-1: DataPoints of geoLocation ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| latitude | xs:float | RW | false | deg | The current latitude part of a geo-location. |
| longitude | xs:float | RW | false | deg | The current longitude part of a geo-location. |
| altitude | xs:float | RW | true | m | The optional current altitude part of a geo-location. |
| heading | xs:float | RW | true |  | The azimuth of a device measured in degrees to true north. North is 0.0 degrees, east is 90.0 degrees, south is 180.0 degrees, west is 270.0 degrees. A negative value indicates an unknown heading. |
| horizontalAccuracy | xs:float | R | true |  | The optional current horizontal accuracy of the geo-location. The unit of measures is meters and describes a radius around the latitude/longitude coordinate. |
| verticalAccuracy | xs:float | R | true | m | The optional current vertical accuracy of the altitude. |
| headingAccuracy | xs:float | R | true | deg | The optional current maximum deviation between the heading and the true geomagnetic heading. |
| targetLatitude | xs:float | RW | true | deg | The optional target latitude part of a geo-location. This can be used to move a device to a new location. |
| targetLongitude | xs:float | RW | true | deg | The optional target longitude part of a geo-location. This can be used to move a device to a new location. |
| targetAltitude | xs:float | RW | true | m | The optional target altitude part of a geo-location. This can be used to move a device to a new altitude. |

#### glucometer

This ModuleClass provides the capability to report the measurement of glucose characteristics.

Table 5.3.1.42-1: DataPoints of glucometer ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| concentration | xs:float | R | false | mg/dl | The measurement of concentration by Glucometer. |
| hba1c | xs:float | R | true | pct | The measurement of HbA1c by Glucometer. |
| contextExercise | xs:float | R | true | pct | The measurement of context exercise by Glucometer. |
| contextMedication | xs:float | R | true | mg/dl | The measurement of context medication by Glucometer. |
| contextCarbohydratesAmount | xs:float | R | true | g | The measurement of context carbohydrates by Glucometer. |
| contextCarbohydratesSource | xs:string | R | true |  | The timing of meals (for example “breakfast carbohydrates”). |
| contextMeal | xs:string | R | true |  | The style of meals (for examople “casual”). |
| contextLocation | xs:string | R | true |  | The body location where the Glucometer is worn (for example “finger”). |
| contextTester | xs:string | R | true |  | The test style (for example “self”). |
| contextHealth | xs:string | R | true |  | The severity of symptoms (for example “minor”), |

#### grinder

This ModuleClass is for controlling a grinder, for example in a coffee machine.

Table 5.3.1.43-1: DataPoints of grinder ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| useGrinder | xs:boolean | RW | false |  | The current status of the grinder enablement. "True" indicates enabled, and "False" indicates disabled. |
| coarseness | hd:enumGrindCoarseness | RW | true |  | The wished coarseness of the solid supplies after grinding, for example for coffee beans (see clause 5.6.21). |
| grainsRemaining | hd:enumGrainsLevel | R | true |  | The level of remaining grains in a machine having a grinder, for example for remaining coffee beans in the coffee machine grinder (see clause 5.6.20). |

#### heatingZone

This ModuleClass provides the capabilities to monitor the status of the heating zone, for example for a cooktop.

Table 5.3.1.44-1: DataPoints of heatingZone ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| heatingLevel | xs:integer | R | false |  | The current heating level of the zone. The value range is from 0 (indicating that the zone is not heating) up to the maxHeatingLevel. |
| maxHeatingLevel | xs:integer | R | false |  | The maximum value allowed for the heating level of the zone. |

#### height

This ModuleClass provides the capability to report the measurement of height.

Table 5.3.1.45-1: DataPoints of height ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| height | xs:float | R | false | cm | The height measurement. |

#### hotWaterSupply

This ModuleClass provides information about the status of supplying hot water into tanks or bath tubs.

Table 5.3.1.46-1: DataPoints of hotWaterSupply ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| bath | xs:boolean | RW | true |  | The status of whether a bath tub is filled. |

#### impactSensor

This ModuleClass describes the capabilities on an impact sensor. The impact is a high force or shock over a short time period and the impactSensor detects this.

Table 5.3.1.47-1: DataPoints of impactSensor ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| impactStatus | xs:boolean | R | false |  | The “impactStatus” indicates as follows:  “True” means that a physical impact is detected, “False” means indicates a normal status (no impact detected). |
| impactLevel | xs:float | R | true |  | The “impactLevel” provides the level of impact. The unit of measure is “g” (G-force). |
| impactDirectionHorizontal | xs:float | R | true |  | The “impactDirection” indicates the horizontal direction where the impact comes from. The value is 0° to 360°. 0 is the front of the sensor and with clockwise increment. |
| impactDirectionVertical | xs:float | R | true |  | The “impactDirection” indicates the vertical direction where the impact comes from. The value is 0° to 360°. 0 is the front of the sensor and with upward increment. |

#### keepWarm

This module allows to control the ‘keep warm’ feature in devices like coffe machines, kettles etc. It allows to keep water warm for a desired time. This ModuleClass inherits from binarySwitch (see clause 5.3.1.12) to store setting for the ‘keep warm’ feature. If the “powerState” data point in a keepWarmSwitch is “True” then the ‘keep warm’ function will be performed just after boiling (or heating) process is finished (otherwise this function will not be applied).

Table 5.3.1.48-1: DataPoints of keepWarm ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| time | xs:integer | RW | true |  | The desired duration of ‘keep water warm’ function. It indicates how long water shall be kept warm, for example after the boiling in a kettle. The value indicates a time expressed in minutes. |
| targetTemperature | xs:float | RW | true | C | Content temperature |

#### keypad

This ModuleClass provides the capability to perform a user defined service through the key-in number. For example, a user can define key 1 as "perform a takeout from a restaurant with combo meal 1". The IoT service provider or user can define the services.

Table 5.3.1.49-1: DataPoints of keypad ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| keyNumber | xs:integer | R | false |  | The number of the pressed key. |

#### liquidLevel

This ModuleClass provides the desired level of water (or other liquid) for an appliance, for example the desired level of milk for a cup of coffee from a coffee machine.

Table 5.3.1.50-1: DataPoints of liquidLevel ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| liquidLevel | hd:enumLiquidLevel | RW | false |  | The desired level of water or other liquid, for example the desired level of milk in a cup of coffee (see clause 5.6.24). |

#### liquidRemaining

This ModuleClass provides the status of water level (or other liquid) for an appliance, for example the level of remaining milk in a coffee machine.

Table 5.3.1.51-1: DataPoints of liquidRemaining ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| liquidRemaining | hd:enumLiquidLevel | R | false |  | The remaining level of water or other liquid in an appliance (see clause 5.6.24). |

#### lock

This ModuleClass provides the function to lock and unlock an object.

Table 5.3.1.52-1: DataPoints of lock ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| lock | xs:boolean | RW | false |  | "True" indicates the object is locked, while "False" indicates the object is not locked. |

#### mediaSelect

This ModuleClass provides capabilities to control and monitor media input and output of device such as TV or SetTopBox.

Table 5.3.1.53-1: DataPoints of mediaSelect ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| mediaID | xs:integer | RW | false |  | The numeric index of the activated media in the supported media sources list “supportedMediaSources”. |
| supportedMediaSources | list of hd:enumSupportedMediaSource | R | false |  | List of supported input or output media for the given device (see clause 5.6.32). |
| mediaName | xs:string | R | true |  | Specifies a pre-defined media input or output. |
| status | xs:boolean | R | true |  | Specifies whether the specific media instance is selected (“True”) or not (“False”). |
| mediaType | hd:enumSupportedMediaSource | R | false |  | Specifies the type of the media (see clause 5.6.32). |

#### motionSensor

This ModuleClass provides the capabilities to indicate the occurrence of motion and raising of an alarm if the triggering criterion is met.

Table 5.3.1.54-1: DataPoints of motionSensor ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| alarm | xs:boolean | R | false |  | The detection of the motion occurrence. |
| silentTime | xs:integer | RW | true | s | The time that a motionSensor restrains from sending an alarm in case continous motions are detected after one alarm is produced. This data point can be used to avoid repeated alarm reports. |
| sensitivity | xs:integer | RW | true |  | The level of the detection accuracy of the motion sensor. This data point can be used to control the number of the report. |

#### numberValue

This ModuleClass provides the capabilities to represent a number. It also has capabilities for controlled increment and decrement a counter. It can be used to present a number-related functionality in a technology where there is only a weak semantic specification of that functionality.

Table 5.3.1.55-1: Actions of numberValue ModuleClass

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Return Type | Name | Argument | Optional | Documentation |
| none | decrementNumberValue | none | true | Decrement the “numberValue” by the value of “step”, down to the value of “minimum”. |
| none | incrementNumberValue | none | true | Increment the "numberValue" by the value of "stepValue", up to the value of "maxValue". |
| none | resetNumberValue | none | true | Reset the “numberValue” to its “defaultValue”. |

Table 5.3.1.55-2: DataPoints of numberValue ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| numberValue | xs:float | RW | false |  | The actual value of the number. |
| minValue | xs:float | RW | true |  | The optional minimum value of the number. The default is the system-specific minimum value for a float value. |
| maxValue | xs:float | RW | true |  | The optional maximum value of the number. The default is the system-specific maximum value for a float value. |
| defaultValue | xs:float | RW | true |  | The optional default value for the number. The default is 0.0 . |
| step | xs:float | RW | true |  | The optional step size for controlled increment and decrement. The default is 1.0 , even when this data point is not implemeneted. |

#### openLevel

This ModuleClass provides the capabilities to control and monitor the open status of an entity, for example a curtain.

Table 5.3.1.56-1: Actions of openLevel ModuleClass

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Return Type | Name | Argument | Optional | Documentation |
| none | open | None | true | Increase the open level by the amount of the “stepValue” up to the “maxLevel”. |
| none | close | None | true | Decrease the open level by the amount of the “stepValue” down to the “minLevel”. |

Table 5.3.1.56-2: DataPoints of openLevel ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| openLevel | xs:integer | RW | false | pct | The rounded percentage of the current open level of entity in the range of [0, 100]. 0 percentage shall mean the entity is closed. |
| stepValue | xs:integer | RW | true |  | The step value used by the “open” and “close” actions. |
| minLevel | xs:integer | RW | true |  | The minimum value allowed for the “openLevel” status. The default value is 0, which means fully closed. |
| maxLevel | xs:integer | RW | true |  | The maximum value allowed for the “openLevel” status. The default value is 100, which means fully opened. |

#### operationMode

This ModuleClasses provides capabilities to control or monitor the operation mode of appliances.

Table 5.3.1.57-1: DataPoints of operationMode ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| startPause | xs:boolean | RW | false |  | A value of “True” triggers or starts an operation, and “False” pauses the operation. |

#### overcurrentSensor

This ModuleClass provides capabilities for an over-current sensor.

Table 5.3.1.58-1: DataPoints of overcurrentSensor ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| overcurrentStatus | xs:boolean | R | false |  | The overcurrentStatus indicates as follows:  “True” indicates that an over-current is detected, and “False” indicates a normal status, this means that an over-current is not detected. |
| detectedTime | m2m:timestamp | R | true |  | The time when the over-current was detected. |
| duration | xs:float | R | true | ms | The duration of the detected over-current. |

#### oximeter

This ModuleClass provides the capability to report the measurement of blood oxygen characteristics.

Table 5.3.1.59-1: DataPoints of oximeter ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| oxygenSaturation | xs:integer | R | false | pct | The measurement of oxygensaturation by Oximeter. |

#### ozoneMeter

This ModuleClass provides capabilities for an ozone meter. The “ozoneValue…” attributes are optional, but one of them SHALL be provided.

Table 5.3.1.60-1: DataPoints of ozoneMeter ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| ozoneValuePPM | xs:float | R | true | ppm |  |
| ozoneValueMG | xs:float | R | true | *mg/m* |  |
| ozoneStatus | hd:enumOzoneStatus | R | true |  | The ozoneStatus indicates the level of ozone status. (see clause 5.6.26). |
| maxValue | xs:float | R | true |  | The mazimum value shows the measurement range of the ozone meter (for example maxValue=5 means the range is 0 to 5 *ppm*). This attribute is only used that the ozoneMeter provides “ppm” value. |

#### magneticSensorParameters

This ModuleClass provides capabilities to set service parameters for parking detectors.

Table 5.3.1.61-1: DataPoints of magneticSensorParameters ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| detectionInterval | xs:integer | RW | false | s | Time interval for detecting the geomagnetic field strength. |
| initialMagnetic | xs:float | RW | true | [G](https://hansungackr-my.sharepoint.com/personal/andyhan_hansung_ac_kr/Documents/work/0.수행과제/AppData/Local/Microsoft/Windows/c00183154/AppData/Local/Youdao/Dict/Application/7.5.2.0/resultui/dict/) | The value of initial geomagnetic field strength, which is usually set when the parking detector is initially installed. It may be reset later if the geomagnetic field strength of the local environment is affected by other factors. The value is used as the reference threshold to determine the *parkingStatus*. When the parking detector detects that the strength of the magnetic field is greater than the *initialMagnetic*, the *parkingStatus* is set to “true”, otherwise, the *parkingStatus* is set to “false”. |
| magneticSensitivityLevel | xs:integer | RW | true |  | The level of detection sensitivity. It’s implementation specific. |
| highMagneticAlarm | xs:boolean | R | false |  | The alarm of high magnetic interference. The alarm threshold is implementation specific. |

#### parkingStatus

This ModuleClass provides the status of the parking detector.

Table 5.3.1.62-1: DataPoints of parkingStatus ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| parkingStatus | xs:boolean | R | false |  | This value indicates the status of the parking space.   * ”False”: not occupied. * “True”: occupied. |

#### periodicalReportConfig

This ModuleClass provides capabilities to set parameters of periodic report.

Table 5.3.1.63-1: DataPoints of periodicalReportConfig ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| backoffTime | xs:integer | RW | false | s | The number of seconds to wait before connecting to network. |
| period | xs:integer | RW | true | s | Reporting period. |
| retryTimes | xs:integer | RW | true |  | The maximum number of re-sent attempts in the case of report failure. |
| retryInterval | xs:integer | RW | true | s | The minimum time interval between each message transmission retry. |

#### phoneCall

This ModuleClass provides the capability get or set the caller and receipient IDs as well as to initate and terminate a call.

Table 5.3.1.64-1: Actions of phoneCall ModuleClass

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Return Type | Name | Argument | Optional | Documentation |
| none | call | none | true | Initiate an outgoing call. |
| none | answer | none | true | Answer (pickup) an incoming call. |
| none | hangup | none | true | Hangup an established call. |

Table 5.3.1.64-2: DataPoints of phoneCall ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| callerID | xs:string | RW | false |  | This data point represents the initiating caller identiﬁcation of a call. In case of an outgoing call this would be the local line ID. This data point is optional. When it is empty for an incomming call, then the caller ID is unknown. When it is empty for an outgoing call, then it is exepected that it is set by the PBX or the operator. The format of caller ID is not speciﬁed here. |
| recipientID | xs:string | RW | false |  | This data point represents the receiving caller identiﬁcation of a call. In case of an incoming call this would be the local line ID and optionally extension. The format of caller ID is not speciﬁed here. |
| callState | hd:enumCallState | R | true |  | This data point represents the current state of an associated phone device regarding calls. |

#### playerControl

This ModuleClass provides capabilities to control and monitor the operational modes of a media player functionality.

Table 5.3.1.65-1: Actions of playerControl ModuleClass

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Return Type | Name | Argument | Optional | Documentation |
| none | nextTrack | none | true | Go forward to a next chapter, section or similar marker in the media. |
| none | previousTrack | none | true | Go back to a previous chapter, section or similar marker in the media. |

Table 5.3.1.65-2: DataPoints of playerControl ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| currentPlayerMode | hd:enumPlayerMode | RW | false |  | The current mode of the player. |
| currentPlayerModeName | xs:string | R | true |  | Name of current player mode in string. This can be used when “currentPlayerMode” is vendor-specific. |
| supportedPlayerModes | list of hd:enumPlayerMode | R | false |  | List of supported modes for a player. |
| speedFactor | xs:float | RW | true |  | The optional factor of speeding up or slowing down playback, rewind or fast forward. |

#### powerSave

This ModuleClass provides capabilities to enable the power saving mode of a device and monitor the current status.

Table 5.3.1.66-1: DataPoints of powerSave ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| powerSaveEnabled | xs:boolean | RW | false |  | The current status of the power saving mode. "True" indicates enabled, and "False" indicates not enabled. |

#### printQueue

This ModuleClass provides the capabilities for monitoring printing list information.

Table 5.3.1.67-1: DataPoints of printQueue ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| uri | list of  xs:uri | trueR | false |  | The URI of the printing file. The URI could be “file://www.example.com/file.extension” |
| printingState | list of hd:enumJobState | R | false |  | The printingState is indicating the status of the printing file. |

#### pulsemeter

This ModuleClass provides the capability to report the measurement of pulse characteristics.

Table 5.3.1.68-1: DataPoints of pulsemeter ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R-W | Optional | Unit | Documentation |
| pulseRate | xs:float | R | false | bpm | The measurement of pulserate by pulsemeter. |
| rr | xs:float | R | true | ms | The measurement of RR interval by pulsemeter. |
| energy | xs:float | R | true | kcal/h | The measurement of energy by pulsemeter. |
| modality | xs:string | R | true |  | The modality of a particular SpO2 measurement. |

#### pushButton

This ModuleClass provides the capability to indicate the operation of a push button style switch. A typical application can be an SOS button.

Table 5.3.1.69-1: DataPoints of pushButton ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| pushed | xs:boolean | R | false |  | This data point indicates the press of the button. |

#### recorder

This ModuleClass provides the capability to record video/audio for a defined duration.

Table 5.3.1.70-1: DataPoints of recorder ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| duration | xs:integer | RW | false | s | The duration for video/audio recording. Set to trigger the recorder. |

#### refrigeration

This ModuleClass provides capabilities for a refrigeration function.

Table 5.3.1.71-1: DataPoints of refrigeration ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| rapidFreeze | xs:boolean | RW | true |  | Controls the rapid freeze capability. “True” indicates active, “False” indicates inactive. |
| rapidCool | xs:boolean | RW | true |  | Controls the rapid cool capability. “True” indicates active, “False” indicates inactive. |
| defrost | xs:boolean | RW | true |  | Controls the defrost cycle. “True” indicates active, “False” indicates inactive. |
| deodorize | xs:boolean | RW | true |  | Controls the deodorize cycle. “True” indicates active, “False” indicates inactive. |
| degerm | xs:boolean | RW | true |  | Controls the degerm cycle. “True” indicates active, “False” indicates inactive. |

#### relativeHumidity

This ModuleClass provides the capability for a device to report the humidity based on a specified rule that is vendor dependent.

Table 5.3.1.72-1: DataPoints of relativeHumidity ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| relativeHumidity | xs:float | R | false | pct | The measurement of the relative humidity value |
| desiredHumidity | xs:float | RW | true | pct | Desired value for humidity. This data point indicates the desired humidity. |

#### remoteControlEnable

This ModuleClasses provides capabilities to monitor the remote controllability of the appliance.

Table 5.3.1.73-1: DataPoints of remoteControlEnable ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R-W | Optional | Unit | Documentation |
| remoteControlEnabled | xs:boolean | R | false |  | This data point enables or disables remote controllability and is set by a user locally. “True” indicates enabled remote access, and “False” indicates disabled remote access. |

#### robotCleanerJobMode

This ModuleClasses provides capabilities to control and monitor the job modes of a robotCleaner.

Table 5.3.1.74-1: DataPoints of robotCleanerJobMode ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| currentJobMode | hd:enumRobotCleanerJobMode | RW | false |  | Currently active job mode (see clause 5.6.28). |
| currentJobModeName | xs:string | R | true |  | Name of current job mode as a string. This can be used when “currentJobMode” is vendor-specific. |
| jobModes | list of hd:enumRobotCleanerJobMode | R | false |  | List of possible job states the device supports (see clause 5.6.28). |

#### 

#### runState

This ModuleClasses provides capabilities to control and the monitor machine state of appliances.

Table 5.3.1.75-1: DataPoints of runState ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| currentMachineState | hd:enumMachineState | RW | false |  | The currently active machine state (see clause 5.6.25). |
| machineStates | list of hd:enumMachineState | R | false |  | A list of possible machine states the device supports (see clause 5.6.25). |
| currentJobState | hd:enumJobState | R | true |  | The currently active job state at the level of some transaction being executed by the device (see clause 5.6.23). |
| jobStates | list of hd:enumJobState | R | true |  | The list of possible job states that the device supports (see clause 5.6.23). |
| progressPercentage | xs:float | R | true | pct | The indication of current job progress in percentage. |

#### securityMode

This ModuleClasses provides capabilities to control and monitor a security mode.

Table 5.3.1.76-1: DataPoints of securityMode ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| currentSecurityMode | hd:enumSecurityMode | RW | false |  | Current security mode (see clause 5.6.29). |
| securityModes | list of hd:enumSecurityMode | R | false |  | List of possible security modes the device supports (see clause 5.6.29). |

#### sessionDescription

This ModuleClass provides the capabilities for a sessionDescription containing a URL at twhich the specified media can be accessed and the definition of media using SDP.

Table 5.3.1.77-1: DataPoints of mediaType ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| url | xs:uri | R | true |  | A URL at which the specified media can be accessed. |
| sdp | xs:string | R | true |  | Media description using SDP. One or more comma separated multiple SDP lines (SDP media or attribute line) can be included using SDP description syntax as defined in the SDP specification in RFC4566 [16]. |

#### signalStrength

This ModuleClass provides the capability to monitor the strength of the signal.

Table 5.3.1.78-1: DataPoints of signalStrength ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| lqi | xs:integer | R | false | pct | The current value of link quality indicator, which reflects the scaling of rssi by dividing the received signal strength over reference signal strength. The common unit for lqi is percentage [0,100]. For the detailed definition, please see IEEE 802.15.4 [i.4], clause 6.7.8. |
| rssi | xs:float | R | true |  | The current value of received signal strength indicator, which reflects the raw signal level. |

Note: Need further study for this module class.

#### slcAlarm

This ModuleClass provides capabilities to provide alarm information of street light controller.

Table 5.3.1.79-1: DataPoints of slcAlarm ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| overCurrentThreshold | xs:float | RW | true | A | The threshold of over current. |
| overCurrentDuration | xs:integer | RW | true | min | The duration of over current to trigger the alarm. |
| underCurrentThreshold | xs:float | RW | true | A | The threshold of under current. |
| underCurrentDuration | xs:integer | RW | true | min | The duration of under current to trigger the alarm. |
| overVoltageThreshold | xs:float | RW | true | V | The threshold of over voltage. |
| overVoltageDuration | xs:integer | RW | true | min | The duration of over voltage to trigger the alarm. |
| underVoltageThreshold | xs:float | RW | true | V | The threshold of under voltage. |
| underVoltageDuration | xs:integer | RW | true | min | The duration of under voltage to trigger the alarm. |
| overVoltagePercent | xs:integer | RW | true | pct | The threshold of over current in terms of percentage. The value range is [0,100]. This is the alternative trigger of the *overVoltageAlarm* and should be mutually exclusive to the *overVoltageThreshold*. |
| underVoltagePercent | xs:integer | RW | true | pct | The duration of under current in terms of percentage. The value range is [0,100]. This is the alternative trigger of the *underVoltageAlarm* and should be mutually exclusive to the *underVoltageThreshold*. |
| standardreferenceVoltage | xs:float | RW | true | V | The reference voltage used as the basis of the *overVoltagePercent* and *underVoltagePercent*. |
| overCurrentAlarm | xs:boolean | R | false |  | The alarm of over current. |
| underCurrentAlarm | xs:boolean | R | false |  | The alarm of under current. |
| overVoltageAlarm | xs:boolean | R | false |  | The alarm of over voltage. |
| underVoltageAlarm | xs:boolean | R | false |  | The alarm of under voltage. |

#### slcParameterSetting

This ModuleClass provides capabilities to set service parameters.

Table 5.3.1.80-1: DataPoints of slcParameterSetting ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| lightCount | xs:integer | RW | true |  | Number of lampholders controlled by the street light controller. |
| timePlanStatus | xs:boolean | RW | true |  | “False” indicates the time plan is not used. “True” indicates the time plan is being used. |
| timeRangeCount | xs:integer | RW | true |  | The number of time ranges for the time plan. |
| timeRange | list of xs:time | RW | true |  | A array of sequencial time points which define the time plan. Each time point is the start time of the next time range as well as the end of previous time range in the time plan. |
| timeRangeLightDimmingValue | list of xs:string | RW | true |  | A array containing the dimming values in different time ranges. In the case that *lightCount* is larger than 1, it is a 2-dimentional array describing the dimming value of each lampholder in each time range.  Editor’s Note: the data type is FFS for this datapoint. |

#### slcReportInfo

This ModuleClass provides information of status of the street light controller.

Table 5.3.1.81-1: DataPoints of slcReportInfo ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| voltageFrequency | xs:float | R | true | Hz | The frequency of voltage. |
| switchStatus | xs:boolean | RW | false |  | This value indicates the status of light switch. ”False”: off, “True”: on. |
| onlineStatus | xs:boolean | R | true |  | This value indicates the communication status of light. ”False”: offline, “True”: online. |
| lightVoltage | xs:float | R | false | V | The voltage of the street light controller. |
| lightCurrent | xs:float | R | false | A | The current of the street light controller. |
| lightDimmingValue | xs:integer | RW | false | pct | The dimming value of the street light. |
| lightPowerFactor | xs:float | R | false |  | The light power factor is the ratio of active power to apparent power. Power factor is a factor that measures the efficiency of electrical equipment. The value range is [0,1]. |
| lightActivePower | xs:float | R | true | kW | Active power is the electrical power required to keep the electrical equipment running normally, that is, the electrical power that converts electrical energy into other forms of energy, such as mechanical, optical, thermal, and so on. |
| lightReactivePower | xs:float | R | true | kVA | The reactive power is the electrical power required to establish an alternating magnetic field and induced magnetic flux. |
| lightApparentPower | xs:float | R | true | kVA | This value indicates the apparent power that is mainly used to calculate the energy consumption of the street light. |
| lightPolarizationAxis | xs:float | R | true | degree | The angle of the polarization axis in case street light devices support this feature. The PolarizationAxis is measured starting from Editors Note: DESCRIPTION NEEDED |
| colourTemperature | xs:integer | R | true | K | The current colour temperature of the street lights. |
| lampTechnology | xs:string | R | true |  | A string that indicates the type of lamp technology that is used in the street lamps, e.g. “LED”, “Tungsten”, etc |

#### smokeSensor

This ModuleClass provides the capabilities to indicate the detection of smoke and raising an alarm if the triggering criterion is met.

Table 5.3.1.82-1:: Actions of smokeSensor ModuleClass

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Return Type | Name | Argument | Optional | Documentation |
| none | muteSmokeSensor | none | true | Mute the smoke sensor alarm. |
| none | test | none | true | Testing the alarm |

Table 5.3.1.82-2: DataPoints of smokeSensor ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R-W | Optional | Unit | Documentation |
| alarm | xs:boolean | R | false |  | The alarm is indicated as follows:  “True” indicates that smoke has been detected, “False” indicates a normal status, that means that smoke is not detected. |
| detectedTime | m2m:timestamp | RW | true |  | The date and time the smoke is detected. |
| smokeThreshhold | xs:integer | RW | true | ppm | The threshhold to trigger the alarm. |
| currentValue | xs:integer | R | true |  | The current data value of the smoke sensor. |
| sensorFault | xs:boolean | R | true |  | “True” indicates the sensor fault status of smoke sensor. “False” indicates the sensor fault of smoke sensor has beenis eliminated. |
| lowVoltage | xs:boolean | R | true |  | “True” indicates the low voltage status of smoke sensor. “False” indicates the low voltage alarm of smoke sensor has beenis eliminated. |
| dismantled | xs:boolean | R | true |  | “True” indicates the smoke sensor is dismantled. “False” indicates the dismantled alarm of smoke sensor has beenis eliminated. |
| powerOn | xs:boolean | R | true |  | “True” indicates the smoke sensor is powered on. “False” is invalid. |

#### sphygmomanometer

This ModuleClass provides the capability to report the measurement of blood pressure characteristics.

Table 5.3.1.83-1: DataPoints of sphygmomanometer ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R-W | Optional | Unit | Documentation |
| diastolicPressure | xs:float | R | false | mmHg | The measurement of diastolic pressure by sphygmomanometer. |
| systolicPressure | xs:float | R | false | mmHg | The measurement of systolic pressure by sphygmomanometer. |
| meanPressure | xs:float | R | false | mmHg | The measurement of mean arterial pressure by sphygmomanometer. |

#### spinLevel

This ModuleClass provides capabilities to control and monitor the level of spin. It is intended to be part of devices which use spinning function such as a washing machine and a dryer.

Table 5.3.1.84-1: DataPoints of spinLevel ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| spinLevelStrength | hd:enumSpinLevelStrength | RW | false |  | The value of spin-dry level (see clause 5.6.30). A higher value indicates a higher spin level. |

#### steamClosetJobMode

This ModuleClasses provides capabilities to control and monitor the job modes of steamCloset.

Table 5.3.1.85-1: DataPoints of steamClosetJobMode ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| currentJobMode | hd:enumSteamClosetJobMode | RW | false |  | Currently active job mode (see clause 5.6.31). |
| currentJobModeName | xs:string | R | true |  | Name of current job mode as a string. This can be used when “currentJobMode” is vendor-specific. |
| jobModes | list of hd:enumSteamClosetJobMode | R | false |  | List of possible job states the device supports (see clause 5.6.31). |

#### televisionChannel

This ModuleClass provides capabilities to set and get channels of a device that has a channel list.

Table 5.3.1.86-1: Actions of televisionChannel ModuleClass

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Return Type | Name | Argument | Optional | Documentation |
| none | upChannel | None | true | Change the current channel to the next channel in the stored list of available channels. If the current channel is the last one in the list, the new set channel may be the first one in the list. |
| none | downChannel | None | true | Change the current channel to the previous channel in the stored list of available channels. If the current channel is the first one in the list, the new set channel may be the last one in the list. |

Table 5.3.1.86-2: DataPoints of televisionChannel ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| channelNumber | xs:integer | RW | false |  | Current channel number. |
| availableChannels | list of xs:integer | R | true |  | The list of available channel numbers which may be build by automatic scan and/or manual selction. |
| previousChannel | xs:integer | R | true |  | The channel number which was selected previously. |
| channelName | xs:string | R | true |  | Current human-friendly channel name in string, for example ‘CNN’. |

#### temperature

This ModuleClass provides capabilities to represent the current temperature and target temperature of devices such as an air conditioner, refrigerator, oven etc.

Table 5.3.1.87-1: DataPoints of temperature ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| currentTemperature | xs:float | R | false |  | The current temperature. |
| targetTemperature | xs:float | RW | true |  | The desired temperature to reach. |
| unit | hd:enumTemperatureUnit | RW | true | C or F or K | Default values is ‘C’ |
| minValue | xs:float | R | true |  | Minimum value of “targetTemperature”. |
| maxValue | xs:float | R | true |  | Maximum value of “targetTemperature”. |
| stepValue | xs:float | R | true |  | Step value allowed for “targetTemperature”. |

#### temperatureAlarm

This ModuleClass provides the capabilities to indicate the detection of abnormal temperatures and raises an alarm if the triggering criterion is met.

Table 5.3.1.88-1: DataPoints of temperatureAlarm ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
|  |  |  |  |  |  |
| unit | hd:enumTemperatureUnit | RW | true | C or F or K | Default value is ‘C’ |
| temperature | xs:float | R | true | Defined in the datapoint ‘unit’ | To report the value of the temperature. |
|  |  |  |  |  |  |
| highTemperatureAlarm | xs:boolean | R | false |  | High temperature alarm |
| highTemperatureAlarmThreshold | xs:float | RW | true | Defined in the datapoint ‘unit’ | The threshold of maximum temperature alarm. |
| lowTemperatureAlarm | xs:boolean | R | false |  | Low temperature alarm |
| lowTemperatureAlarmThreshold | xs:float | RW | true | Defined in the datapoint ‘unit’ | The threshold of minimum temperature alarm. |
| alarmTimestamp | xs:datetime | R | true |  | The timestamp since the alarm is active |

#### textMessage

This ModuleClass provides capabilities to set and get a text message.

Table 5.3.1.89-1: Actions of textMessage ModuleClass

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Return Type | Name | Argument | Optional | Documentation |
| none | resetTextMessage | none | true | Reset the receiver of the message to the “defaultValue”. |

Table 5.3.1.89-2: DataPoints of textMessage ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| textMessage | xs:string | RW | false |  | The current message value. |
| supportedMessageValues | list of xs:string | R | true |  | List of supported values for the message. Each of the values in this list must be URL-encoded. An encoded value must not contain white spaces. |
| minLength | xs:integer | R | true |  | The optional minimum length in characters of the message. The default is 0. |
| maxLength | xs:integer | R | true |  | The optional maximum length in characters of the message. The default is unlimited. |
| messageEncoding | xs:string | R | true |  | The optional expected method for character encoding of the message. The default is "UTF-8". |
| defaultValue | xs:string | RW | true |  | The optional default value for “textMessage”. The default is an empty string. |

#### timer

This ModuleClass provides capabilities to monitor and control the times when the appliance executes its operations, that means when it starts, when it ends etc.

Table 5.3.1.90-1: Actions of timer ModuleClass

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Return Type | Name | Argument | Optional | Documentation |
| none | activateClockTimer | None | true | Activate current clock timer. |
| none | deactivateClockTimer | None | true | Deactivate current clock timer. |

Table 5.3.1.90-2: DataPoints of timer ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| referenceTimer | xs:integer | R | true |  | A timer (for example. a time-based value, App Defined Epoch, Progressive) expressed in seconds. The value indicates a time counter to be used as reference for the other time-based data points of this ModuleClass. Usually it is the time since the last event of power-on of the producer (or more in detail the time since the boot of its connectivity node). |
| targetTimeToStart | xs:integer | RW | true |  | A time span (for example a time-based value, App Defined Epoch, Fixed) expressed in seconds. The value indicates the time when the appliance is expected to start its operation, starting counting from the last ”referenceTimer”. |
| targetTimeToStop | xs:integer | RW | true |  | A time span (for example a time-based value, App Defined Epoch, Fixed) expressed in seconds. The value indicates the time when the appliance is expected to stop its operation, starting counting from the last ”referenceTimer”. |
| estimatedTimeToEnd | xs:integer | R | true |  | A timer (for example a time-based value, App Defined Epoch, Progressive) expressed in seconds. The value indicates the time to the end of an appliance’s operations. It is calculated at runtime by the device itself during the execution of its operation. |
| runningTime | xs:integer | R | true |  | A timer (for example a time-based value, App Defined Epoch, Progressive) expressed in seconds. It indicates the time of the current operation. Usually its value is increasing one value per second. It starts counting from 0 when the operation starts and stops counting when the operation ends. |
| targetDuration | xs:integer | R | true |  | A time span (for a time-based value, App Defined Epoch, Fixed) expressed in seconds. The value indicates a time that represents the target duration of the operation as per user selection. |
| absoluteStartTime | m2m:timestamp | RW | true |  | An absolute time to specify the start time. |
| absoluteStopTime | m2m:timestamp | RW | true |  | An absolute time to specify the stop time. |

#### turbo

This ModuleClass provides capabilities to enable turbo mode and monitor the current status of the turbo function. It is intended to be part of devices which use turbo function such as an air conditioner, a washing machine etc.

Table 5.3.1.91-1: DataPoints of turbo ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| turboEnabled | xs:boolean | RW | false |  | The current status of the turbo mode. "True" indicates enabaled, and "False" indicates not enabled. |

#### uvSensor

This ModuleClass describes the capabilities of an ultraviolet sensor.

Table 5.3.1.92-1: DataPoints of uvSensor ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| uvValue | xs:float | R | false | mW/cm2 |  |
| uvStatus | hd:enumUvStatus | R | true |  | The “uvStatus” indicates the level of the UV radiation status (see clause 5.6.35). |

#### waterFlow

This ModuleClass provides capabilities for controlling the water strength of a device.

Table 5.3.1.93-1: DataPoints of waterFlow ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| waterLevelStrength | hd:enumWaterFlowStrength | RW | false |  | The desired level of water flow (see clause 5.6.37). A higher value indicates higher water flow. |

#### waterMeterAlarm

This ModuleClass provides capabilities to provide alarm information of watermeter, such as the alarm of low water flow. Once an alarm is fired, a notification should be sent out from the device and no historical alarm is stored locally, therefore no need to associate timestamp with the alarms.

Table 5.3.1.94-1: DataPoints of waterMeterAlarm ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| highFlowAlarmThreshold | xs:float | RW | false | m3 | The threshold of continuous high flow alarm. |
| highFlowDuration | xs:integer | RW | true | s | The duration of high water flow. |
| highFlowAlarm | xs:boolean | R | false |  | Continuous high water flow alarm |
| lowFlowAlarmThreshold | xs:float | RW | true | m3 | The threshold of continuous low flow alarm. |
| lowFlowDuration | xs:integer | RW | true | s | The duration of low water flow. |
| lowFlowAlarm | xs:boolean | R | true |  | Continuous low water flow alarm |
| reverseFlowAlarmThreshold | xs:float | RW | true | m3 | The threshold of continuous reverse flow alarm. |
| reverseFlowDuration | xs:integer | RW | true | s | The duration of reverse water flow. |
| reverseFlowAlarm | xs:boolean | R | false |  | Reverse flow alarm |
| highPressureAlarmThreshold | xs:float | RW | false | bar | The threshold of high water pressure alarm. |
| highPressureAlarm | xs:boolean | R | false |  | High water pressure alarm |
| lowPressureAlarmThreshold | xs:float | RW | false | bar | The threshold of low water pressure alarm. |
| lowPressureAlarm | xs:boolean | R | false |  | Low water pressure alarm |
| highTemperatureAlarmThreshold | xs:float | RW | false | C | The threshold of maximum water temperature alarm. |
| highTemperatureAlarm | xs:boolean | R | false |  | High water temperature alarm |
| lowTemperatureAlarmThreshold | xs:float | RW | false | C | The threshold of minimum water temperature alarm. |
| lowTemperatureAlarm | xs:boolean | R | false |  | Low water temperature alarm |
| highTemperatureInnerAlarmThreshold | xs:float | RW | true | C | The threshold of high temperature alarm inside water meter. |
| innerHighInternalTemperatureAlarm | xs:boolean | R | true |  | Internal high temperature alarm |
| innerErrorAlarm | xs:boolean | R | true |  | Internal error alarm |
| innerTemperatureSensorFault | xs:boolean | R | true |  | Internal temperature sensor failure |
| tamperAlarm | xs:boolean | R | true |  | Data was tampered alarm |
| waterTemperatureSensorFault | xs:boolean | R | false |  | Water temperature sensor failure |
| pressureSensorFault | xs:boolean | R | true |  | Pressure sensor failure |
| communicationAlarm | xs:boolean | R | true |  | Communication abnormality alarm |
| magneticInterference | xs:boolean | R | true |  | Magnetic interference warning |
| storageFault | xs:boolean | R | true |  | Storage failure alarm |
| urgencyButtonPush | xs:boolean | R | true |  | Indicate the event of the urgency button being pushed. Pushing the urgency button may give a user a temporary right to still use the water meter (e.g. for 3 days) after an unpaid bill. |
| buttonFault | xs:boolean | R | true |  | Button error flag |
| demolitionAlarm | xs:boolean | R | true |  | Demolition sign |
| impulseFault | xs:boolean | R | true |  | Pulse anomaly flag |
| vibrationSensorFault | xs:boolean | R | true |  | Vibration sensor failure |

#### waterMeterReportInfo

This ModuleClass provides information of measurements of the watermeter.

Table 5.3.1.95-1: DataPoints of waterMeterReportInfo ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| dailyUseWaterTime | xs:integer | R | true | s | The total time of water consumption daily. |
| cumulativeFlow | xs:float | R | false | m3 | The total consumption of water(since the activation of the meter). |
| cumulativeFlowDaily | xs:float | R | true | m3 | The cumulative daily consumption of water (begins at 00:00). |
| positiveCumulativeFlow | xs:float | R | true | m3 | The positive cumulative consumption of water daily (begins at 00:00). |
| negativeCumulativeFlow | xs:float | R | true | m3 | The negative cumulative consumption of water daily (begins at 00:00). |
| peakFlowRate | xs:float | R | true | m3/h | Daily maximum instantaneous water flow rate. |
| peakFlowRateTime | m2m:timestamp | R | true |  | The timestamp of the daily highest instantaneous water flow rate. |
| lowestFlowRate | xs:float | R | true | m3/h | Daily lowest instantaneous water flow rate. |
| lowestFlowRateTime | m2m:timestamp | R | true |  | The timestamp of daily lowest instantaneous water flow rate. |
| peakReverseFlowRate | xs:float | R | true | m3/h | Daily reverse maximum instantaneous water flow rate. |
| peakReverseFlowRateTime | m2m:timestamp | R | true |  | The timestamp of daily reverse highest instantaneous water flow rate. |
| lowestReverseFlowRate | xs:float | R | true | m3/h | Daily reverse lowest instantaneous water flow rate. |
| lowestReverseFlowRateTime | m2m:timestamp | R | true |  | The timestamp of daily reverse lowest instantaneous water flow rate. |
| intervalFlow | list of xs:float | R | true | m3 | Water consumption records measured at the interval of “flowInterval” describerd in 5.3.1.96 per day. |
| reverseIntervalFlow | list of xs:float | R | true | m3 | Water reverse consumption records measured at the interval of “reverseFlowInterval” described in 5.3.1.96 per day. |
| waterIntervalTemperature | list of xs:float | R | true | C | Water temperature records measured at the interval of “waterTemperatureInterval” described in 5.3.1.96 per day. |
| waterIntervalPressure | list of xs:float | R | true | bar | Water pressure records measured at the interval of “waterPressureInterval” described in 5.3.1.96 per day. |

#### waterMeterSetting

This ModuleClass provides capabilities to set service parameters for data sampling and reporting.

Table 5.3.1.96-1: DataPoints of waterMeterSetting ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| flowInterval | xs:integer | RW | false | s | The measurement interval of water consumption. |
| reverseFlowInterval | xs:integer | RW | false | s | The measurement interval of reverse water consumption. |
| waterTemperatureInterval | xs:integer | RW | true | s | The measurement interval of water temperature. |
| waterPressureInterval | xs:integer | RW | true | s | The measurement interval of reverse water pressure. |
| intensiveSampleInterval | xs:integer | RW | true | s | The time interval of intensive data sampling. |
| intensiveReportInterval | xs:integer | RW | true | s | The time interval of intensive data report. |
| intensiveReportStartTime | m2m:timestamp | RW | true |  | The start time of data intensive report. |

#### waterSensor

This ModuleClass provides the capabilities to indicate whether or not water has been sensed, and raising an alarm if the triggering criterion is met.

Table 5.3.1.97-1: DataPoints of waterSensor ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| alarm | xs:boolean | R | false |  | The detection of water.  The alarm is indicated as follows:  “True” indicates that water has been detected, “False” indicates a normal status, that means that water is not detected. |

#### waterQualityMonitor

This ModuleClass provides the information of water quality detection.

Table 5.3.1.98-1: DataPoints of waterQualityMonitor ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| algae | xs:float | R | true | mg/L | Blue-green algae |
| anionics | xs:float | R | true | mg/L | An-ionic surfactant |
| aroh | xs:float | R | true | mg/L | Volatile phenol (ArOH) |
| as | xs:float | R | true | mg/L | Arsenic (As) |
| biotoxicity | xs:float | R | true | mg/L | Biological toxicity |
| bod | xs:float | R | true | mg/L | Biochemical oxygen demand (BOD) is the amount of dissolved oxygen consumed during the biochemical reaction of biodegradable organic matter that is decomposed by microorganisms in water under certain conditions. |
| cod | xs:float | R | true | mg/L | Chemical oxygen demand (COD) is the amount of reducing substance that needs to be oxidized in a water sample. |
| conductivity | xs:float | R | true | S/m  (siemens per meter) | Conductivity is a parameter used to describe the ease of charge flow in a substance. |
| cd | xs:float | R | true | mg/L | Cadmium (Cd) |
| chlorophyll\_a | xs:float | R | true | mg/L | Chlorophyll a |
| cn | xs:float | R | true | mg/L | Cyanide (CN) |
| cr6 | xs:float | R | true | mg/L | Hexavalent chromium (Cr6) |
| cu | xs:float | R | true | mg/L | Cuprum (Cu) |
| do | xs:float | R | true | mg/L | Dissolved oxygen (DO). Molecular oxygen in the air dissolved in water is called dissolved oxygen |
| f | xs:float | R | true | mg/L | Fluoride (F) |
| fe | xs:float | R | true | mg/L | Total iron (Fe) |
| ftu | xs:float | R | true | mg/L | Turbidity (FTU) which refers to degree of hindrance of the solution as the light passes through it. |
| hg | xs:float | R | true | mg/L | Mercury (Hg) |
| kmno4 | xs:float | R | true | mg/L | Permanganate index which refers to the amount of oxidant consumed in the water samples using potassium permanganate as an oxidant in an acidic or alkaline medium. (KMnO4) |
| nh3nh4 | xs:float | R | true | mg/L | Ammonia nitrogen (NH3NH4) is the nitrogen in the form of free ammonia (NH3) and ammonium ions (NH4+) in water. |
| no3n | xs:float | R | true | mg/L | Nitrate nitrogen (NO3N) |
| oil | xs:float | R | true | mg/L | Petroleum pollutants |
| pb | xs:float | R | true | mg/L | Lead (Pb) |
| ph | xs:float | R | true |  | Potential Of Hydrogen (pH) |
| sulfide | xs:float | R | true | mg/L | Sulfide |
| temperature | xs:float | R | true | C | Water temperature |
| tn | xs:float | R | true | mg/L | Total nitrogen (TN) which is defined as the total amount of various forms of inorganic and organic nitrogen in water. |
| tp | xs:float | R | true | mg/L | Total phosphorus (TP) which is the result of the conversion of various forms of phosphorus into orthophosphate after digestion of the water sample, measured in milligrams of phosphorus per liter of water sample. |
| zn | xs:float | R | true | mg/L | Zinc (Zn) |

#### weight

This ModuleClass provides the capability to report the measurement of weight.

Table 5.3.1.99-1: DataPoints of weight ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| weight | xs:float | R | false | kg | The weight measurement. |
| unit | hd:enumWeightUnit | RW | true |  | The unit of measure for the weight values. The default is kilogram (kg). (see clause 5.6.39). |

#### anemometer

This ModuleClass provides the capabilities to indicate the measure of the wind speed.

Table 5.3.1.100-1: DataPoints of anemometer ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| speed | xs:float | R | false | km/h | The speed of the wind |

#### barometer

This ModuleClass provides the capabilities to measure the atmospheric pressure and indicate the detection of abnormal pressure, and raise an alarm if a triggering criterion is met.

Table 5.3.1.101-1: DataPoints of barometer ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| alarm | xs:boolean | R | true |  | This data point indicates the status of detection of an abnormal pressure. “True” indicates an abnormal pressure, “False” indicates a normal pressure. |
| atmosphericPressure | xs:float | R | false | hPa | To report the value of the atmospheric pressure. |
| minPressureThreshold | xs:integer | RW | true | hPa | The min threshold to trigger the alarm. |
| maxPressureThreshold | xs:integer | RW | true | hPa | The max threshold to trigger the alarm. |

#### rainGauge

This ModuleClass provides the capabilities to measure the height of fallen rain.

Table 5.3.1.102-1: Actions of rainGauge ModuleClass

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Return Type | Name | Argument | Optional | Documentation |
| none | reset | none | false | Empty the water container. Set the height value to 0. |

Table 5.3.1.102-2: DataPoints of rainGauge ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| height | xs:integer | R | false | mm | This data point indicates the number of mm of rainfall since the last reset of the device. |

#### infraredSensor

This ModuleClass provides the capabilities to indicate whether or not an object has been sensed.

Table 5.3.1.103-1: DataPoints of infraredSensor ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| detectedValue | xs:boolean | R | false |  | The "detectedValue" indicates as follows:  "True" means that an object is detected, "False" means no object is detected. |
| detectedTime | m2m:timestamp | R | false |  | The time when the object was detected. |

#### disposal

This ModuleClass provides capabilities to control the status of the disposing functionality for garbage disposal.

Table 5.3.1.104-1: DataPoints of disposal ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| disposalStatus | xs:boolean | RW | false |  | The status of disposal. “True” indicates disposing, “False” indicates not disposing. |

#### waterFilterType

This ModuleClass indicates the type of the water purifier.

Table 5.3.1.105-1: DataPoints of waterFilterType ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| filterType | hd:enumWaterFilterType | R | false |  | The type of water purifier. list of Water Filter Type(see clause5.6.41). |

#### touchScreen

This ModuleClass provides the capability to get selections of a user from the pre-defined menus on the screen as parts of a process of charging transportation payment card of the user.

Table 5.3.1.106-1: DataPoints of touchScreen ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| pushed | xs:boolean | R | false |  | This data point indicates the press of the button. |
| positionX | xs:integer | R | false | pixel | This data point indicates the horizontal position of the touching. (1..N) |
| positionY | xs:integer | R | false | pixel | This data point indicates the vertical position of the touching. (1..N) |

#### prePaidCardReader

The Pre-paid card reader ModuleClass provides functions to read NFC card and indicates its information..

Table 5.3.1.107-1: DataPoints of prePaidCardReader ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| readStatus | xs:boolean | R | false |  | This data point indicates the status of reading the pre-paid card.  “True” means the reader reads the pre-paid card successfully. |
| cardInfo | xs:string | R | false |  | The card information is a string |
| balance | xs:float | RW | false |  | This data point indicates the balance of the pre-paid card. |
| currency | xs:string | R | true |  | The currency is depended on the country which the card is used.  For example, the unit could be “Dollar” or “$” in US, “Euro” in EU and “Won” in Korea. |

#### billDeposit

This ModuleClass provides the capability to deposit bills, indicates the balance of the deposited bills and detects fake.

Table 5.3.1.108-1: DataPoints of billDeposit ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| depositStatus | xs:boolean | R | false |  | This data point indicates the deposit is made successfully or not. |
| balance | xs:float | R | false |  | This data point indicates the balance of the deposited bills. |
| count | xs:integer | R | false |  | The data point indicates the number of bills which are deposited. |
| currency | xs:string | R | true |  | The currency is depended on the country which the deposited bills are used.  For example, the currency could be “Dollar” in US, “Euro” in EU and “Won” in Korea. |
| fakeStatus | xs:boolean | R | false |  | This data point indicates that the deposited bills are fake notes. |

#### billWithdrawal

This ModuleClass provides the capability to withdraw bills which are deposited by the billDepositModule.

Table 5.3.1.109-1: DataPoints of billWithdrawal ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| withdrawalStatus | xs:boolean | R | false |  | This data point indicates the withdrawal is made successfully or not. |
| balance | xs:float | R | false |  | This data point indicates the balance of the withdrawn bills. |
| count | xs:integer | R | false |  | The data point indicates the number of bills which are withdrawn. |

#### coinDeposit

This ModuleClass provides the capability to deposit coins, indicates the balance of the deposited coins and detects fake.

Table 5.3.1.110-1: DataPoints of coinDeposit ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| depositStatus | xs:boolean | R | false |  | This data point indicates the deposit is made successfully or not. |
| balance | xs:float | R | false |  | This data point indicates the balance of the deposited coins. |
| count | xs:integer | R | false |  | The data point indicates the number of coins which are deposited. |
| currency | xs:string | R | true |  | The currency is depended on the country that the deposited coins are used.  For example, the unit could be “Dollar” in US, “Euro” in EU and “Won” in Korea. |
| fakeStatus | xs:boolean | R | false |  | This data point indicates that the deposited coins are fake. |

#### cashDispenser

This ModuleClass provides the capability to withdraw designated amount of cash or returns all the deposited bills and coins by the billDepositModule and coinDepositModule.

Table 5.3.1.111-1: DataPoints of cashDispenser ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| dispenseStatus | xs:boolean | R | false |  | This data point indicates the dispense is successful or not. |
| balance | xs:float | R | false |  | This data point indicates the balance of the dispensed bills and coins. |
| count | xs:integer | R | false |  | The data point indicates the number of bills and coins which are dispensed. |
| currency | xs:string | R | true |  | The currency is depended on the country that the dispensed bills and coins are used.  For example, the unit could be “Dollar” in US, “Euro” in EU and “Won” in Korea. |

#### cardScanner

This ModuleClass provides the capability to scan an image of a card, gets the card information from the image and provides the information..

Table 5.3.1.112-1: DataPoints of cardScanner ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| scanStatus | xs:boolean | R | false |  | This data point indicates the scanning process is successful. |
| cardInfo | xs:string | R | false |  | The card information is a string value. |

#### headingSensor

This ModuleClass provides the capabilities to indicate the heading (e.g. status of get-in and get-out) of a pedestrian or a vehicle which crossing a control point (e.g. entrance and gate).

Table 5.3.1.113-1: DataPoints of headingSensor ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| heading | xs:float | RW | false | degree | The azimuth of a device measured in degrees to true north. North is 0.0 degrees, east is 90.0 degrees, south is 180.0 degrees, west is 270.0 degrees. A negative value indicates an unknown heading. |
| headingAccuracy | xs:float | R | true | degree | The optional current maximum deviation between the heading and the true geomagnetic heading. |

#### signalPanel

This ModuleClass displays a signal (e.g. direction arrow) to indicate a gate is permitted to get-in or get-out on a panel.

Table 5.3.1.114-1: DataPoints of signalPanel ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| signal | xs:boolean | RW | false |  | The signal is true, the panel indicates go forward signal or icon. If the signal is false, the panel displays stop signal or icon. |

#### crossingBarrier

An active barrier is used to block a pedestrian or vehicle from a control point (e.g. entrance and gate).

Table 5.3.1.115-1: DataPoints of crossingBarrier ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| barrierDefault | xs:boolean | RW | true |  | The value of the barrier default is true, the default status of the barrier is closed and vice versa. |
| barrierStatus | xs:boolean | RW | false |  | The value of the barrier status is true, the barrier is closed and vice versa. |
| timer | xs:time | RW | true |  | The timer indicates the duration of barrierStatus is changed. This means that barrierStatus is set to the current value of barrierDefault. |

#### threeDDisplay

This ModuleClass provides capabilities to give the information of a 3D display.

Table 5.3.1.116-1: DataPoints of threeDDisplay ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| threeDDisplayType | hd:enum3DDisplayType | R | false |  | The type of 3D display technology (see clause 5.6.52) |
| threeDDisplayViewAngle | xs:integer | R | true | degree | This data point indicates viewing angle of the 3D display (1..360) |
| threeDDisplayResolutionX | xs:integer | R | true | pixel | This data point indicates resolution of X-axis of the 3D display. |
| threeDDisplayResolutionY | xs:integer | R | true | pixel | This data point indicates resolution of Y-axis of the 3D display. |
| threeDGlasses | xs:boolean | R | true |  | This data point indicates the 3D display uses 3D glasses (TRUE) or not. |

#### threeDScanner

This ModuleClass provides the capability to scanning 3D object for the user.

Table 5.3.1.117-1: DataPoints of threeDScanner ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| threeDScannerType | hd:enum3DScannerType | R | false |  | This data point indicates type of the 3D scanner |
| threeDScanResolution | xs:integer | R | false | dpi | This data point indicates the scanning resolution. |
| maxSizeX | xs:float | R | true | cm | This data point indicates the maximum horizontal size of the scanning. |
| maxSizeY | xs:float | R | true | cm | This data point indicates the maximum vertical size of the scanning. |
| maxSizeZ | xs:float | R | true | cm | This data point indicates the maximum depth of the scanning. |

#### blender

This ModuleClass is for controlling a blender, for example in a juicer.

Table 5.3.1.118-1: DataPoints of blender ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| useBlender | xs:boolean | RW | false |  | The current status of the blender enablement. "True" indicates enabled, and "False" indicates disabled. |
| spinSpeed | hd:enumSpinLevelStrength | RW | true |  | Choose the right spin speed according to the hardness of the object (eg.fruits and vegetables) (see clause 5.6.30) |

#### shoesWasherJobMode

This ModuleClasses provides capabilities to control and monitor the job modes of a shoesWasher.

Table 5.3.1.119-1: DataPoints: DataPoints of shoesWasherJobMode ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| currentJobMode | hd: enumShoesWasherJobMode | RW | false |  | The currently active job mode. (see clause5.6.54) |
| currentJobModeName | xs:string | R | true |  | The name of the current job mode as a string. This can be used when the currentJobMode is vendor-specific. |
| jobModes | list of hd: enumShoesWasherJobMode | R | false |  | List of possible job states the device supports (see clause5.6.54) |

### City Domain

### Health Domain

### Home Domain

### Industry Domain

### Vehicular Domain

### Agriculture Domain

#### cowActivityMonitor

This ModuleClass provides capabilities to measure activity data in terms of step count. The data sampling rate is every hour by default.

Table 5.3.7.1-1: DataPoints of cowActivityMonitor ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| stepsPerPeriod | list of xs:integer | R | false |  | Counted steps per hour during each report period. The value is updated every report period. The length of the report period can be configured by the *periodicalReportConfig* ModuleClass. If not configured, the default length of the period is 1 hour.  If the report period is longer than 1 hour, this data point may contain multiple data samples (one for each hour) during last period. The report period should always be set to longer than 1 hour which is the data sampling period. The start time of the data sampling and report period is implementation specific, e.g. the power-on time. |
| updateTime | xs:datetime | R | true |  | A timestamp that indicates the update time of the *stepsPerPeriod* data point. |
| historyStepCounts | list of xs:integer | R | false |  | The list of stepCounts per hour during the last 24 hours (in total 24 data samples). |

### Railway Domain

#### baliseTransmissionModule

A balise is an electronic beacon or transponder placed between the rails of a railway as part of an automatic train protection (ATP) system.

The Balise Transmission Module(BTM) ModuleClasses provides capabilities to indicate and to get balise information.

Table 5.3.8.1-1: DataPoints of baliseTransmissionModule ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| baliseSystemId | hd:enumBaliseSystemIndicator | R | true |  | Get the hd:enumBaliseSystemIndicator (see clause 5.6.38). |
| telegramMessage | xs:string | R | true |  | The telegramMessage is system-specific. |

### Metadata Domain

#### features

This ModuleClass provides metadata information on the parent device.

Table 5.3.9.1-1: DataPoints of features ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| metadata | list of xs:string | RW | true |  | Free metadata information. |
| precision | xs:integer | R | true | pct | Approximated precision of the device data. |
| reliability | xs:integer | R | true | pct | Approximated reliability of the device data. |
| weight | xs:float | R | true | g | The weight of the device. |
| size | list of xs:float | R | true | cm | The size of the device [length, width, height]. |

#### location

This ModuleClass provides information on the location of the parent device.

Table 5.3.9.2-1: DataPoints of location ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| postalAddress | xs:string | RW | true |  | Postal address |
| geoJSON | xs:string | R | true |  | Coordinates in GeoJSON format |
| friendlyLocation | xs:string | RW | true |  | Friendly location (e.g. ‘kitchen’) |
| room | xs:string | RW | true |  | Room ID in a building (e.g. ‘A101’) |

#### localization

This ModuleClass provides capabilities for localizing friendly names.

Table 5.3.9.3-1: DataPoints of localization ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| locale | xs:string | R | false |  | Code in ISO 639-1 |
| friendlyName | xs:string | RW | true |  | Friendly name for the parent device, in the given locale. |
| description | xs:string | RW | true |  | Friendly description for the parent device, in the given locale. |
| friendlyLocation | xs:string | RW | true |  | Friendly location, in the given locale (e.g. ‘cuisine’ for locale ‘fr’). |

#### origin

This ModuleClass provides information on the origin of the parent device data.

Table 5.3.9.4-1: DataPoints of origin ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Documentation |
| originID | xs:string | R | true |  | ID of the original data |
| dataType | xs:string | R | true |  | Data type of the original data |
| dataSourceID | xs:string | R | true |  | ID of the data source that created the original data |

## SubDevice models

### Common Domain

#### subDeviceCuff

A cuff is a subDevice that expresses the attachment device for measuring blood pressure.

Table 5.4.1.1-1: Modules of subDeviceCuff model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 0..1 | See clause 5.3.1.12. |
| sphygmomanometer | sphygmomanometer | 1 | See clause 5.3.1.83. |
| pulsemeter | pulsemeter | 1 | See clause 5.3.1.68. |

#### subDevicePowerOutlet

A powerOutlet is a subDevice that specifies the attachment device for deviceSmartPlug.

Table 5.4.1.2-1: Modules of subDevicePowerOutlet model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 0..1 | See clause 5.3.1.12. |
| energyConsumption | energyConsumption | 0..1 | See clause 5.3.1.32. |
| overcurrentSensor | overcurrentSensor | 0..1 | See clause 5.3.1.58. |
| dimmingLevel | numberValue | 0..1 | See clause 5.3.1.55.  This provides the capability to change the energy. |

### City Domain

### Health Domain

### Home Domain

### Industry Domain

### Vehicular Domain

## Device models

### Common Domain

#### device3DPrinter

A 3D printer is a smart home appliance to provide 3D printing capabilities.

Table 5.5.1.1-1: Modules of device3DPrinter Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| faultDetection | faultDetection | 0..1 | See clause 5.3.1.34. |
| threeDprinter | threeDprinter | 1 | See clause 5.3.1.1. |
| runState | runState | 1 | See clause 5.3.1.75. |
| temperature | temperature | 1 | See clause 5.3.1.87.  This value of “currentTemperature” in this module instance represents the temperature of the nozzle. This value SHALL be a float number in a range from 0.0 to 1000.0. |
| printQueue | printQueue | 1 | See clause 5.3.1.67. |

#### deviceAirQualityMonitor

An air quality monitor is an environmental monitoring device used to monitor the air quality.

Table 5.5.1.2-1: Modules of deviceAirQualityMonitor Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| airQualitySensor | airQualitySensor | 1 | See clause 5.3.1.6. |
| battery | battery | 0..1 | See clause 5.3.1.10. |
| connectivity | connectivity | 0..1 | See clause 5.3.1.25. |
| periodicalReportConfig | periodicalReportConfig | 0..1 | See clause 5.3.1.63. |

#### deviceAudioReceiver

An audio receiver is a device that receives audio signals from a number of sources, processing them to drive speakers.

Table 5.5.3.1-1: Modules of deviceAudioReceiver model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| audioVolume | audioVolume | 1 | See clause 5.3.1.8. |
| mediaInput | mediaSelect | 0..1 | See clause 5.3.1.53. |
| mediaOutput | mediaSelect | 0..1 | See clause 5.3.1.53. |

#### deviceCamera

A camera is an optical instrument for recording or capturing images, which may be stored locally or transmitted to another locations.

Table 5.5.1.4-1: Modules of deviceCamera Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| sessionDescription | sessionDescription | 1 | See clause 5.3.1.77. |
| playerControl | playerControl | 0..1 | See clause 5.3.1.65. |
| motionSensor | motionSensor | 0..1 | See clause 5.3.1.54. |

#### deviceDoor

A door is a device that is used to open and close a door.

Table 5.5.1.5-1: Modules of deviceDoor model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| openLevel | openLevel | 0..1 | See clause 5.3.1.56. |
| doorLock | lock | 0..1 | See clause 5.3.1.52. |
| doorStatus | doorStatus | 0..1 | See clause 5.3.1.29. |

#### deviceDoorLock

A door lock is a device that can be used to lock, for example, a door.

Table 5.5.1.6-1: Modules of deviceDoorLock Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| doorLock | lock | 1 | See clause 5.3.1.52. |
| doorStatus | doorStatus | 0..1 | See clause 5.3.1.29. |
| battery | battery | 0..1 | See clause 5.3.1.10. |

#### deviceLight

A light is a device that is used to control the state of an illumination appliance.

Table 5.5.1.7-1: Modules of deviceLight Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| faultDetection | faultDetection | 0..1 | See clause 5.3.1.34. |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| runState | runState | 0..1 | See clause 5.3.1.75. |
| colour | colour | 0..1 | See clause 5.3.1.23. |
| colourSaturation | colourSaturation | 0..1 | See clause 5.3.1.24. |
| brightness | brightness | 0..1 | See clause 5.3.1.17. |
| timer | timer | 0..1 | See clause 5.3.1.90. |

#### deviceMultiFunctionPrinter

A Multi Function Printer (MFP) is an office machine which incorporates the functionality of multiple devices in one, so as to have a smaller footprint in home or office. A typical MFP may act as a combination of printer, scanner and more. This MFP information model provides capabilities to control and monitor MFP specific functions and resources.

Table 5.5.1.8-1: Modules of deviceMultiFunctionPrinter Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| printerRunState | runState | 1 | See clause 5.3.1.75. |
| scannerRunState | runState | 0..1 | See clause 5.3.1.75. |
| autoDocumentFeeder | autoDocumentFeeder | 0..1 | See clause 5.3.1.9. |
| printQueue | printQueue | 0..1 | See clause 5.3.1.67. |

#### devicePrinter

A printer is a device that is used to monitor or control the state of a printing appliance.

Table 5.5.1.9-1: Modules of devicePrinter Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| runState | runState | 0..1 | See clause 5.3.1.75. |
| printQueue | printQueue | 0..1 | See clause 5.3.1.67. |

#### deviceScanner

A scanner is a device that optically scans images, printed text, handwriting or an object, and converts it to a digital image.

Table 5.5.1.10-1: Modules of deviceScanner model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| runState | runState | 1 | See clause 5.3.1.75. |
| autoDocumentFeeder | autoDocumentFeeder | 0..1 | See clause 5.3.1.9. |

#### deviceSmartPlug

A smart plug is a device that can turn on and off a connected appliance.

Table 5.5.1.11-1: Modules of deviceSmartPlug model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 0..1 | See clause 5.3.1.12. |
| faultDetection | faultDetection | 0..1 | See clause 5.3.1.34. |
| remoteControlEnable | remoteControlEnable | 0..1 | See clause 5.3.1.73. |

Table 5.5.1.11-2: Subdevice of deviceSmartPlug Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Subdevice Instance Name | Subdevice Name | Multiplicity | Description |
| powerOutlet0 | subDevicePowerOutlet | 1..N | See clause 5.4.1.2. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

#### deviceSwitch

A switch is a device that is used to control and monitor the state of power.

Table 5.5.1.12-1deviceSwitch Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |

#### deviceThermometer

A thermometer is a device that can be used to check, for example, the body or other temperatures.

Table 5.5.1.13-1: Modules of deviceThermoMeter Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| temperature | temperature | 1 | See clause 5.3.1.87. |
| battery | battery | 1 | See clause 5.3.1.10. |
| temperatureAlarm | temperatureAlarm | 0..1 | See clause 5.3.1.88 |

#### deviceThermostat

A thermostat is used to control the ambient temperature of rooms within, for example, a house. This information model provides capabilities to interact with specific functions of thermostats.

Table 5.5.1.14-1: Modules of deviceThermostat Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| runState | runState | 0..1 | See clause 5.3.1.75.  The possible values of the “supportedModes” datapoint for the thermostat device are included in clause 5.6.23. |
| timer | timer | 0..1 | See clause 5.3.1.90. |
| temperature | temperature | 1 | See clause 5.3.1.87. |

#### deviceWaterValve

A water valve is a device that is used to turn the water supply ON or OFF remotely.

Table 5.5.1.15-1: Modules of waterValve Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| openLevel | openLevel | 1 | See clause 5.3.1.56. |

#### deviceServiceButton

A service button is a device that initiates and controls technical or business processes, such as ordering consumer and industrial goods over the Internet. It may support optional ModuleClasses to present information to a user, such as presenting textual and graphical information.

Table 5.5.1.16-1: Modules of deviceServiceButton Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| pushButton | pushButton | 1 | See clause 5.3.1.69. |
| connectivity | connectivity | 0..1 | See clause 5.3.1.25 |
| operationMode | operationMode | 0..1 | See clause 5.3.1.57. |
| battery | battery | 0..1 | See clause 5.3.1.10. |
| textMessage | textMessage | 0..1 | See clause 5.3.1.89. This ModuleClass may be used to present textual information to a user. |
| image | binaryObject | 0..1 | See clause 5.3.1.11. This ModuleClass may be used to present graphical information to a user. |
| credentials | credentials | 0..1 | See clause 5.3.1.27. |
| geoLocation | geoLocation | 0..1 | See clause 5.3.1.41. This ModuleClass may be used to detect and report the geo-location of a deviceServiceButton device. |

#### deviceGenericSensor

A device that is composed of one or more basic sensors. This generic model is proposed to represent very simple appliances that feature one or more sensing behaviors (mono/multi sensors).

Table 5.5.1.17-1: Modules of deviceGenericSensor Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Optional | Description |
| temperatureAlarm | temperatureAlarm | 0..1 | See clause 5.3.1.88 |
| acousticSensor | acousticSensor | 0..1 | See clause 5.3.1.2 |
| impactSensor | impactSensor | 0..1 | See clause 5.3.1.47 |
| motionSensor | motionSensor | 0..1 | See clause 5.3.1.54 |
| smokeSensor | smokeSensor | 0..1 | See clause 5.3.1.82 |
| uvSensor | uvSensor | 0..1 | See clause 5.3.1.92 |
| waterSensor | waterSensor | 0..1 | See clause 5.3.1.84 |

#### device3DDisplay

A 3D display is a device to display 3D contents such as 3D character or holographic image.

Table 5.5.1.18-1: Modules of device3DDisplay Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| clock | clock | 0..1 | See clause 5.3.1.18. |
| audioVolume | audioVolume | 0..1 | See clause 5.3.1.8. |
| threeDDisplay | threeDDisplay | 1 | See clause 5.3.1.116. |
| connectivity | connectivity | 1 | See clause 5.3.1.25. |
| machineState | runState | 1 | See clause 5.3.1.75. |
| lock | lock | 0..1 | See clause 5.3.1.52. |

#### device3DScanner

A 3D scanner is a device to scan 3D objects such as a statue of a character.

Table 5.5.1.19-1: Modules of device3DScanner device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| clock | clock | 0..1 | See clause 5.3.1.18. |
| threeDScanState | runState | 1 | See clause 5.3.1.75. |
| threeDScanner | threeDScanner | 1 | See clause 5.3.1.117. |
| connectivity | connectivity | 1 | See clause 5.3.1.25. |

### City Domain

#### deviceOutdoorLamp

An outdoor lamp is a smart home appliance to provide lights and information for outside of home with smart sensing capabilities such as ultraviolet sensing.

Table 5.5.2.1-1: Modules of deviceOutdoorLamp Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| brightness | brightness | 0..1 | See clause 5.3.1.17. |
| motionSensor | motionSensor | 0..1 | See clause 5.3.1.54. |
| airQualitySensor | airQualitySensor | 0..1 | See clause 5.3.1.6. |
| uvSensor | uvSensor | 0..1 | See clause 5.3.1.92. |
| timer | timer | 0..1 | See clause 5.3.1.90.  The timer is used to set duration of giving lights from the moment of triggering by the “brightness” module or “motionSensor” module. |
| faultDetection | faultDetection | 0..1 | See clause 5.3.1.34. |

#### deviceMagneticParkingMonitor

A parking detector is used to detect the state of the parking space. When the vehicle enters the parking space, the detector measure the change of the geomagnetic field strength, and report the state of the parking space to a server.

Table 5.5.2.2-1: Modules of deviceMagneticParkingMonitor Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| battery | battery | 0..1 | See clause 5.3.1.10. |
| connectivity | connectivity | 0..1 | See clause 5.3.1.25. |
| periodicalReportConfig | periodicalReportConfig | 0..1 | See clause 5.3.1.63. |
| ParkingStatus | ParkingStatus | 1 | See clause 5.3.1.62. |
| magneticSensorParameters | magneticSensorParameter | 1 | See clause 5.3.1.61. |

#### deviceSmartElectricMeter

A smart electric meter is a metering device that is used to measure consumption data for electrictricity.

Table 5.5.2.3-1: Modules of deviceSmartElectricMeter Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| faultDetection | faultDetection | 0..1 | See clause 5.3.1.34. |
| battery | battery | 0..1 | See clause 5.3.1.10. |
| binarySwitch | binarySwitch | 0..1 | See clause 5.3.1.12. |
| runState | runState | 0..1 | See clause 5.3.1.75. |
| clock | clock | 0..1 | See clause 5.3.1.18. |
| energyConsumption | energyConsumption | 1 | See clause 5.3.1.32. |
| energyGeneration | energyGeneration | 0..1 | See clause 5.3.1.33. |

#### deviceSmartGasMeter

A smart gas meter is a metering device that is used to measure consumption data for gas.

Table 5.5.2.4-1: Modules of deviceSmartGasMeter Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| battery | battery | 0..1 | See clause 5.3.1.10. |
| faultDetection | faultDetection | 0..1 | See clause 5.3.1.34. |
| connectivity | connectivity | 0..1 | See clause 5.3.1.25. |
| periodicalReportConfig | periodicalReportConfig | 0..1 | See clause 5.3.1.63. |
| gasMeterReportInfo | gasMeterReportInfo | 1 | See clause 5.3.1.40. |
| gasMeterControl | binarySwitch | 0..1 | See clause 5.3.1.12. |
| gasMeterAlarm | gasMeterAlarm | 0..1 | See clause 5.3.1.39. |

#### deviceSmartWaterMeter

A smart water meter is a metering device that is used to measure consumption data for water.

Table 5.5.2.5-1: Modules of deviceSmartWaterMeter Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| battery | battery | 0..1 | See clause 5.3.1.10. |
| faultDetection | faultDetection | 0..1 | See clause 5.3.1.34. |
| connectivity | connectivity | 0..1 | See clause 5.3.1.25. |
| periodicalReportConfig | periodicalReportConfig | 0..1 | See clause 5.3.1.63. |
| waterMeterSetting | waterMeterSetting | 0..1 | See clause 5.3.1.96. |
| waterMeterReportInfo | waterMeterReportInfo | 1 | See clause 5.3.1.95. |
| waterMeterControl | binarySwitch | 0..1 | See clause 5.3.1.12. |
| waterMeterAlarm | waterMeterAlarm | 0..1 | See clause 5.3.1.94. |

#### deviceStreetLightController

A street light controller is used to control the opening and closing of the street light.

Table 5.5.2.6-1: Modules of deviceStreetLightController Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| geoLocation | geoLocation | 0..1 | See clause 5.3.1.41. |
| connectivity | connectivity | 0..1 | See clause 5.3.1.25. |
| periodicalReportConfig | periodicalReportConfig | 0..1 | See clause 5.3.1.63. |
| slcReportInfo | slcReportInfo | 1 | See clause 5.3.1.81. |
| slcParameterSetting | slcParameterSetting | 1 | See clause 5.3.1.80. |
| slcAlarm | slcAlarm | 1 | See clause 5.3.1.79. |

#### deviceWaterQualityMonitor

An water quality monitor is an environmental monitoring device used to monitor water quality.

Table 5.5.2.7-1: Modules of deviceWaterQualityMonitor Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| battery | battery | 0..1 | See clause 5.3.1.10. |
| connectivity | connectivity | 0..1 | See clause 5.3.1.25. |
| periodicalReportConfig | periodicalReportConfig | 0..1 | See clause 5.3.1.63. |
| waterQualityMonitor | waterQualityMonitor | 1 | See clause 5.3.1.98. |

#### deviceWeatherStation

A weather station is a device that measures various atmospheric parameters.

Table 5.5.2.8-1: Modules of deviceWeatherStation Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| outdoorTemperature | temperature | 1 | See clause 5.3.1.87 |
| indoorTemperature | temperature | 0..1 | See clause 5.3.1.87 |
| airQualitySensor | airQualitySensor | 0..1 | See clause 5.3.1.6 |
| anemometer | anemometer | 0..1 | See clause 5.3.1.100 |
| barometer | barometer | 0..1 | See clause 5.3.1.101 |
| rainGauge | rainGauge | 0..1 | See clause 5.3.1.102 |
| acousticSensor | acousticSensor | 0..1 | See clause 5.3.1.2 |
| uvSensor | uvSensor | 0..1 | See clause 5.3.1.92 |

### Health Domain

#### deviceBloodPressureMonitor

A blood pressure monitor is a device that can be used to monitor the blood pressure and is composed of one or more cuffs and a main montor machine.

Table 5.5.3.1-1: Modules of deviceBloodPressureMonitor Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| battery | battery | 1 | See clause 5.3.1.10 |
| binarySwitch | binarySwitch | 0..1 | See clause 5.3.1.12 |

Table 5.5.3.1-2: Subdevice of deviceBloodPressureMonitor Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Subdevice Instance Name | Subdevice Name | Multiplicity | Description |
| cuff | subDeviceCuff | 1..N | See clause 5.4.1.1 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

#### deviceGlucosemeter

A glucometer is a device that can be used to monitor the blood glucose level.

Table 5.5.3.2-1: Modules of deviceGlucoseMeter Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| glucometer | glucometer | 1 | See clause 5.3.1.42. |
| battery | battery | 1 | See clause 5.3.1.10. |

#### deviceHeartRateMonitor

A heart rate monitor is a device that can be used to monitor the heart rate.

Table 5.5.3.3-1: Modules of deviceHeartRateMonitor Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| pulsemeter | pulsemeter | 1 | See clause 5.3.1.68. |
| battery | battery | 1 | See clause 5.3.1.10. |

#### devicePulseOximeter

A pulseoximeter is a device that can be used to monitor the blood characteristics.

Table 5.5.3.4-1: Modules of devicePulseOxiMeter Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| pulsemeter | pulsemeter | 0..1 | See clause 5.3.1.68.  When the “oximeter” module doesn’t exist, then the “pulsemeter” module is mandatory. |
| oximeter | oximeter | 0..1 | See clause 5.3.1.59.  When the “pulsemeter” module doesn’t exist, then the “oximeter” module is mandatory. |
| battery | battery | 1 | See clause 5.3.1.10. |

#### deviceWeightScaleAndBodyCompositionAnalyser

A weight scale and body composition analyser is a device that can be used to monitor the weight and body composition.

Table 5.5.3.5-1: Modules of deviceWeightScaleAndBodyCompositionAnalyser Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| weight | weight | 1 | See clause 5.3.1.99. |
| bodyCompositionAnalyser | bodyCompositionAnalyser | 1 | See clause 5.3.1.14. |
| bioElectricalImpedanceAnalysis | bioElectricalImpedanceAnalysis | 1 | See clause 5.3.1.13. |
| battery | battery | 1 | See clause 5.3.1.10. |

### Home Domain

#### deviceAirConditioner

An air conditioner is a home appliance used to alter the properties of air (primarily temperature and humidity) to more comfortable conditions. This air conditioner information model provides capabilities to control and monitor air conditioner specific functions and resources.

Table 5.5.4.1-1: Modules of deviceAirConditioner Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| runState | runState | 0..1 | See clause 5.3.1.75. |
| airConJobMode | airConJobMode | 0..1 | See clause 5.3.1.1. |
| airConOperationMode | operationMode | 0..1 | See clause 5.3.1.57.  This module instance is used to trigger an airCon operation that is pre-set in “airConJobMode”. If this data point is not present, then the air conditioner’s job mode can be triggered by setting the “airConJobMode”. |
| airCleanOperationMode | operationMode | 0..1 | See clause 5.3.1.57.  This module instance is used to trigger airClean operation. |
| temperature | temperature | 0..1 | See clause 5.3.1.87. |
| timer | timer | 0..1 | See clause 5.3.1.90. |
| sleepTimer | timer | 0..1 | See clause 5.3.1.90.  The sleep function, which is vendor-specific algorithm (for example increasing the remperature by one degree for every 30 minutes), is triggered instantly when “targetDuration” is set, and it indicates the time to the end of appliance operation. It is set at runtime by a user application. |
| turbo | turbo | 0..1 | See clause 5.3.1.91. |
| airFlow | airFlow | 0..1 | See clause 5.3.1.4. |
| powerSave | powerSave | 0..1 | See clause 5.3.1.66. |
| airQualitySensor | airQualitySensor | 0..1 | See clause 5.3.1.6. |
| filterInfo | filterInfo | 0..1 | See clause 5.3.1.35. |

#### deviceAirPurifier

An airPurifieris a home appliance is used to prevent dust and other particles from air by filtering, washing or electrostatic precipitation. This airPurifier information model provides capabilities to control and monitor airPurifier specific functions and resources.

Table 5.5.4.2-1: Modules of deviceAirPurifier Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| runState | runState | 0..1 | See clause 5.3.1.75. |
| airPurifierJobMode | airPurifierJobMode | 0..1 | See clause 5.3.1.5. |
| airPurifierOperationMode | operationMode | 0..1 | See clause 5.3.1.57.  This module instance is used to trigger the airPurifier operation. |
| timer | timer | 0..1 | See clause 5.3.1.90. |
| powerSave | powerSave | 0..1 | See clause 5.3.1.66. |
| airQualitySensor | airQualitySensor | 0..1 | See clause 5.3.1.6. |
| filterInfo | filterInfo | 0..1 | See clause 5.3.1.35. |

#### deviceClothesDryer

A clothes dryer is a home appliance for drying clothes. This clothesDryer information model provides capabilities to control and monitor clothes dryer specific functions and resources.

Table 5.5.4.3-1: Modules of deviceClothesDryer Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| runState | runState | 0..1 | See clause 5.3.1.75. |
| clothesDryerJobMode | clothesDryerJobMode | 0..1 | See clause 5.3.1.19. |
| clothesDryerOperationMode | operationMode | 0..1 | See clause 5.3.1.57.  This module instance is used to trigger the clothesDryer operation. |

#### deviceClothesWasher

A clothes washer is a home appliance that is used to wash laundry, such as clothing and sheets. This information model provides capabilities to interact with specific functions and resources of clothes washers.

Table 5.5.4.4-1: Modules of deviceClothesWasher Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| clothesWasherOperationMode | operationMode | 0..1 | See clause 5.3.1.57. |
| runState | runState | 1 | See clause 5.3.1.75. |
| clothesWasherJobMode | clothesWasherJobMode | 1 | See clause 5.3.1.20. |
| clothesWasherJobModeOption | clothesWasherJobModeOption | 0..1 | See clause 5.3.1.22. |
| remoteControlEnable | remoteControlEnable | 0..1 | See clause 5.3.1.73. |
| timer | timer | 0..1 | See clause 5.3.1.90. |

#### deviceClothesWasherDryer

A clothes washer dryer is a home appliance that is a combination of cloth washer and cloth dryer in a single cabinet. This information model provides capabilities to interact with specific functions and resources of clothes washers and dryers.

**Table 5.5.4.5-1: Modules of deviceClothesWasherDryer Device model**

|  |  |  |  |
| --- | --- | --- | --- |
| **Module Instance Name** | **Module Class Name** | **Multiplicity** | **Description** |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| clothesWasherDryerOperationMode | operationMode | 0..1 | See clause 5.3.1.57. |
| runState | runState | 1 | See clause 5.3.1.75. |
| clothesWasherDryerJobMode | clothesWasherDryerJobMode | 1 | See clause 5.3.1.21. |
| clothesWasherJobModeOption | clothesWasherJobModeOption | 0..1 | See clause 5.3.1.22. |
| remoteControlEnable | remoteControlEnable | 0..1 | See clause 5.3.1.73. |
| timer | timer | 0..1 | See clause 5.3.1.90. |

#### deviceCoffeeMachine

A coffee machine is a device that is used to brew a coffee, may add foamed milk, and may include some variants, for example a grinder.

Table 5.5.4.6-1: Modules of deviceCoffeeMachine Device model

|  |  |  |  |
| --- | --- | --- | --- |
| **Module Instance Name** | **Module Class Name** | **Multiplicity** | **Description** |
| faultDetection | faultDetection | 0..1 | See clause 5.3.1.34. |
| runState | runState | 0..1 | See clause 5.3.1.75. |
| clock | clock | 0..1 | See clause 5.3.1.18. |
| brewing | brewing | 1 | See clause 5.3.1.16. |
| waterStatus | liquidRemaining | 0..1 | See clause 5.3.1.51. |
| milkStatus | liquidRemaining | 0..1 | See clause 5.3.1.51. |
| grinder | grinder | 0..1 | See clause 5.3.1.43. |
| milkFoaming | foaming | 0..1 | See clause 5.3.1.36. |
| milkQuantity | liquidLevel | 0..1 | See clause 5.3.1.50. |
| brewingSwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| keepWarm | keepWarm | 0..1 | See clause 5.3.1.48. |

#### deviceCookerHood

A cooker hood is a device containing a mechanical fan that hangs above the stove or cooktop in the kitchen.

Table 5.5.4.7-1: Modules of deviceCookerHood model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Optional | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| runState | runState | 0..1 | See clause 5.3.1.75. |
| airFlow | airFlow | 0..1 | See clause 5.3.1.4. |
| cookerHoodJobMode | cookerHoodJobMode | 0..1 | See clause 5.3.1.26. |

#### deviceCooktop

A cooktop is a device that is a kitchen appliance designed for the purpose of cooking food.

Table 5.5.4.8-1: Modules of deviceCooktop model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| heatingZone0 | heatingZone | 1..N | See clause 5.3.1.44. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

#### deviceDehumidifier

A dehumidifier is a device that is used to monitor or control the state of a dehumidifying appliance.

Table 5.5.4.9-1: Modules of deviceDehumidifier Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| relativeHumidity | relativeHumidity | 0..1 | See clause 5.3.1.72. |
| runState | runState | 0..1 | See clause 5.3.1.75. |
| dehumidifierJobMode | dehumidifierJobMode | 0..1 | See clause 5.3.1.28. |
| dehumidifierOperationMode | operationMode | 0..1 | See clause 5.3.1.57.  This module instance is used to trigger dehumidifier operation. |
| Timer | timer | 0..1 | See clause 5.3.1.90. |
| powerSave | powerSave | 0..1 | See clause 5.3.1.66. |

#### deviceDigitalGallery

A digital gallery is a device that is used to display picture, e.g., paintings from artists, photos from photographers or personals etc.

**Table 5.5.4.10-1: Modules of deviceDigitalGallery Device model**

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| galleryMode | galleryMode | 1 | See clause 5.3.1.37 |
| pictureInput | mediaSelect | 1 | See clause 5.3.1.53 |
| powerSave | powerSave | 0..1 | See clause 5.3.1.66 |
| clock | clock | 0..1 | See clause 5.3.1.18 |

#### deviceDishWasher

A dish washer is a home appliance used to wash dishes. This information model provides capabilities to interact with specific functions and resources of a dish washer.

Table 5.5.4.11-1: Modules of deviceDishWasher Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| runState | runState | 0..1 | See clause 5.3.1.75. |
| dishWasherJobMode | dishWasherJobMode | 0..1 | See clause 5.3.1.29. |

#### deviceFan

A fan is a device that is used to monitor or control the state of a fanning device.

Table 5.5.4.12-1: Modules of deviceFan model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |

#### deviceFoodProbe

A food probe is a device that is used to measure the internal temperature of food.

Table 5.5.4.13-1: Modules of deviceFoodProbe model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| temperature | temperature | 1 | See clause 5.3.1.87. |

#### deviceFreezer

A freezer is a large container like a fridge in which the temperature is kept below freezing point, so that food can be storeed inside of it for long periods. This freezer information model provides capabilities to monitor freezer specific functions and resources.

Table 5.5.4.14-1: Modules of deviceFreezer Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| temperature | temperature | 1 | See clause 5.3.1.87. |

#### deviceHomeCCTV

A home CCTV is a smart home appliance to provide monitoring capabilities when people stay way from their home or a room, or to monitor the environmental status of their home or room.

Table 5.5.4.15-1: Modules of deviceHomeCCTV Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| sessionDescription | sessionDescription | 1 | See clause 5.3.1.77. |
| playerControl | playerControl | 0..1 | See clause 5.3.1.65. |
| runState | runState | 0..1 | See clause 5.3.1.75. |
| motionSensor | motionSensor | 0..1 | See clause 5.3.1.54. |
| airQualitySensor | airQualitySensor | 0..1 | See clause 5.3.1.6. |
| ozoneMeter | ozoneMeter | 0..1 | See clause 5.3.1.60. |
| smokeSensor | smokeSensor | 0..1 | See clause 5.3.1.82. |
| acousticSensor | acousticSensor | 0..1 | See clause 5.3.1.1. |
| impactSensor | impactSensor | 0..1 | See clause 5.3.1.47. |
| faultDetection | faultDetection | 0..1 | See clause 5.3.1.34. |
| alarmSpeaker | alarmSpeaker | 0..1 | See clause 5.3.1.7. |

#### deviceHumidifier

A humidifier is a device that is used to monitor or control the state of a humidifying appliance.

Table 5.5.4.16-1: Modules of deviceHumidifier Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |

#### deviceKettle

Kettle is a device used to boil water. It may set a desired temperature for water and may keep water warm for a desired time.

Table 5.5.4.17-1: Modules of deviceKettle Device model

|  |  |  |  |
| --- | --- | --- | --- |
| **Module Instance Name** | **Module Class Name** | **Multiplicity** | **Description** |
| faultDetection | faultDetection | 0..1 | See clause 5.3.1.34. |
| runState | runState | 0..1 | See clause 5.3.1.75. |
| waterStatus | liquidRemaining | 0..1 | See clause 5.3.1.51. |
| boilingSwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| temperature | temperature | 0..1 | See clause 5.3.1.87. |
| keepWarm | keepWarm | 0..1 | See clause 5.3.1.48. |

#### deviceMicrogeneration

A microgeneration is a Home Energy Management System (HEMS) device that is used to generate energy. Examples of microgeneration devices are photovoltaics device or fuel cells.

Table 5.5.4.18-1: Modules of deviceMicrogeneration Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| faultDetection | faultDetection | 0..1 | See clause 5.3.1.34. |
| binarySwitch | binarySwitch | 0..1 | See clause 5.3.1.12. |
| runState | runState | 0..1 | See clause 5.3.1.75. |
| energyGeneration | energyGeneration | 1 | See clause 5.3.1.33. |

#### deviceOven

An oven is a home appliance used to roast and heat food in a complete stove. This information model is applicable to different types of ovens: gas ovens, electrical ovens, steam ovens, microwave ovens, etc. This information model provides capabilities to interact with specific functions and resources of ovens.

Table 5.5.4.19-1: Modules of deviceOven Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12.  Depending on the manufacturer policies or regulations, the binary switch might only be used to turn the device off. |
| runState | runState | 0..1 | See clause 5.3.1.75. |
| timer | timer | 0..1 | See clause 5.3.1.90. |
| temperature | temperature | 1 | See clause 5.3.1.87. |

#### deviceRefrigerator

A refrigerator is a home appliance used to store food at temperatures which are a few degrees above the freezing point of water. This information model provides capabilities to interact with specific functions and resource of refrigerators.

Table 5.5.4.20-1: Modules of deviceRefrigerator Device model

|  |  |  |  |
| --- | --- | --- | --- |
| **Module Instance Name** | **Module Class Name** | **Multiplicity** | **Description** |
| binarySwitch | binarySwitch | 0..1 | See clause 5.3.1.12. |
| powerSave | powerSave | 0..1 | See clause 5.3.1.66. |
| doorStatus | doorStatus | 0..1 | See clause 5.3.1.29. |
| frozenTemperature | temperature | 0..1 | See clause 5.3.1.87. |
| fridgeTemperature | temperature | 1 | See clause 5.3.1.87. |
| customTemperature | temperature | 0..1 | See clause 5.3.1.87.  This module can be configured to fridge temperature or frozen temperature based on its usage by manufacturer |
| refrigeration | refrigeration | 0..1 | See clause 5.3.1.71. |
| controlPanelLock | lock | 0..1 | See clause 5.3.1.52. |
| waterFilterInfo | filterInfo | 0..1 | See clause 5.3.1.35. |

#### deviceRiceCooker

An rice cooker is a home appliance used to cook and heat food. It may set a desired heating time for food and may keep food warm for a desired time.

Table 5.5.4.21-1: Modules of deviceRiceCooker Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12.  Depending on the manufacturer policies or regulations, the binary switch might only be used to turn the device off. |
| faultDetection | faultDetection | 0..1 | See clause 5.3.1.34. |
| runState | runState | 0..1 | See clause 5.3.1.75. |
| timer | timer | 0..1 | See clause 5.3.1.90. |
| keepWarm | keepWarm | 0..1 | See clause 5.3.1.48. |

#### deviceRobotCleaner

A robot cleaner is an autonomous robotic vacuum cleaner that has intelligent programming and a limited vacuum cleaning system. This robot cleaner information model provides capabilities to control and monitor robot cleaner specific functions and resources.

Table 5.5.4.22-1: Modules of deviceRobotCleaner Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| runState | runState | 0..1 | See clause 5.3.1.75. |
| robotCleanerJobMode | robotCleanerJobMode | 1 | See clause 5.3.1.74. |
| robotCleanerOperationMode | operationMode | 0..1 | See clause 5.3.1.57. |
| battery | battery | 0..1 | See clause 5.3.1.10. |
| timer | timer | 0..1 | See clause 5.3.1.90. |
| remoteControlEnable | remoteControlEnable | 0..1 | See clause 5.3.1.73. |

#### deviceSecurityPanel

A security pannel is a device that can change the security mode of, for example, an alarm system.

Table 5.5.4.23-1: Modules of deviceSecurityPanel model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| securityMode | securityMode | 1 | See clause 5.3.1.76. |

#### deviceSetTopBox

A set top box is a device that in general contains a TV tuner input and displays output to a TV.

Table 5.5.4.24-1: Modules of deviceSetTopBox model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| audioVolume | audioVolume | 0..1 | See clause 5.3.1.8. |
| Channel | televisionChannel | 0..1 | See clause 5.3.1.86. |
| mediaInput | mediaSelect | 0..1 | See clause 5.3.1.53. |
| mediaOutput | mediaSelect | 0..1 | See clause 5.3.1.53. |

#### deviceSteamCloset

A deviceSteamCloset is a home appliance that de-wrinkles, sanitizes and dries to clean fabrics similar to a dry cleaner. This information model provides capabilities to interact with specific functions and resources of the steam closet.

Table 5.5.4.25-1: Modules of deviceSteamCloset Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| runState | runState | 0..1 | See clause 5.3.1.75. |
| steamClosetJobMode | steamClosetJobMode | 1 | See clause 5.3.1.85. |
| steamClosetOperationMode | operationMode | 0..1 | See clause 5.3.1.57. |

#### deviceStorageBattery

A storage battery is a Home Energy Management System HEMS device that is used to provide the home with electrical energy.

Table 5.5.4.26-1: Modules of deviceStorageBattery Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| faultDetection | faultDetection | 0..1 | See clause 5.3.1.34. |
| binarySwitch | binarySwitch | 0..1 | See clause 5.3.1.12. |
| runState | runState | 0..1 | See clause 5.3.1.75. |
| battery | battery | 1 | See clause 5.3.1.10. |

#### deviceTelevision

A television (TV) is a home appliance used to show audio and visual content such as broadcasting programs and network streaming. This TV information model provides capabilities to control and monitor TV specific resources.

Table 5.5.4.27-1: Modules of deviceTelevision Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| audioVolume | audioVolume | 0..1 | See clause 5.3.1.8. |
| televisionChannel | televisionChannel | 0..1 | See clause 5.3.1.86. |
| playerControl | playerControl | 0..1 | See clause 5.3.1.65. |
| mediaInput | mediaSelect | 0..1 | See clause 5.3.1.53. |
| mediaOutput | mediaSelect | 0..1 | See clause 5.3.1.53. |

#### deviceWaterHeater

A water heater is a device that is used to provide hot water through home facilities.

Table 5.5.4.28-1: Modules of deviceWaterHeater Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| faultDetection | faultDetection | 0..1 | See clause 5.3.1.34. |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| runState | runState | 0..1 | See clause 5.3.1.75. |
| clock | clock | 0..1 | See clause 5.3.1.18. |
| boiler | boiler | 0..1 | See clause 5.3.1.15. |
| hotWaterSupply | hotWaterSupply | 0..1 | See clause 5.3.1.46. |

#### deviceWindowShade

The window shade is an appliance that provides the ability to cover windows. This device type includes but not limited to roller shades, drapes, and tilt-only blinds.

Table 5.5.4.29-1: Modules of deviceWindowShade Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| openLevel | openLevel | 1 | See clause 5.3.1.56. |
| battery | battery | 0..1 | See clause 5.3.1.10. |
| timer | timer | 0..1 | See clause 5.3.1.90. |

#### deviceBottleWarmer

A bottle warmer is an appliance designed for the purpose of warming the feeding bottle.

Table 5.5.4.30-1: Modules of deviceBottleWarmer Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| keepWarm | keepWarm | 0..1 | See clause 5.3.1.87. |
| runState | runState | 0..1 | See clause 5.3.1.75. |

#### deviceGarbageDisposal

A Garbage Disposal is an appliance designed for the purpose of disposing the kitchen waste.

Table 5.5.4.31-1: Modules of deviceGarbageDisposal Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| runState | runState | 0..1 | See clause 5.3.1.87. |
| disposal | disposal | 1 | See clause 5.3.1.104. |

#### deviceWaterPurifier

A Water Purifier is an appliance to filter the impurity substance in water by different filter elements. This Water Purifier information model provides capabilities to control and monitor Water Purifier specific functions and resources.

Table 5.5.4.32-1: Modules of deviceWaterPurifier Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| runState | runState | 0..1 | See clause 5.3.1.75. |
| timer | timer | 0..1 | See clause 5.3.1.90. |
| waterQualityMonitor | waterQualityMonitor | 0..1 | See clause 5.3.1.90. |
| filterInfo | filterInfo | 0..1 | See clause 5.3.1.35. |
| waterLevel | liquidRemaining | 0..1 | See clause 5.3.1.90. |
| waterFilter | waterFilterType | 0..1 | See clause 5.3.1.90. |

#### deviceAirHeater

An air heater is a home appliance used to increase the temperature of an indoor space, for example, a room. This information model provides capabilities to interact with specific functions and resources of air heaters.

Table 5.5.4.33-1: Modules of deviceAirHeater Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| airHeating | runState | 0..1 | See clause 5.3.1.75.  This module class is used to control and monitor the state of an air heater. |
| temperature | temperature | 1 | See clause 5.3.1.87. |
| timer | timer | 0..1 | See clause 5.3.1.90. |

#### deviceJuicer

Juicer is a machine that can quickly press fruits and vegetables into juice. It may set a desired stir time for juice and can choose the right spin speed of blender according to the hardness of the object (such as fruits and vegetables)  which you want to stir.

Table 5.5.4.y-1: Modules of deviceJuicer Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12.  Depending on the manufacturer policies or regulations, the binary switch might only be used to turn the device off. |
| faultDetection | faultDetection | 0..1 | See clause 5.3.1.34. |
| runState | runState | 0..1 | See clause 5.3.1.75. |
| blender | blender | 0..1 | See clause 5.3.1.118 |
| timer | timer | 0..1 | See clause 5.3.1.90. |

#### deviceShoesWasher

A shoes washer is a home appliance used to wash shoes. This information model provides capabilities to interact with specific functions and resources of a shoes washer.

Table 5.5.4.35-1: Modules of deviceShoesWasher Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| runState | runState | 1 | See clause 5.3.1.75. |
| shoesWasherJobMode | shoesWasherJobMode | 1 | See clause5.3.1.119 |
| timer | timer | 0..1 | See clause 5.3.1.90. |

### Industry Domain

### Vehicular Domain

#### deviceElectricVehicleCharger

An electric vehicle charger is a device that is used for charging or discharging electric vehicles.

Table 5.5.6.1-1: Modules of deviceElectricVehicleCharger Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| faultDetection | faultDetection | 1 | See clause 5.3.1.34. |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| runState | runState | 1 | See clause 5.3.1.75. |
| battery | battery | 1 | See clause 5.3.1.10. |
| electricVehicleConnector | electricVehicleConnector | 1 | See clause 5.3.1.31. |

#### deviceElectricMotorcycle

An electric motorcycle is a plug-in electric vehicle with two wheels. The electricity is stored on board in a rechargeable battery which drives one or more electric motors. This information model provides capabilities to interact with specific functions and resources of an electric motorcycle.

Table 5.5.6.2-1: Modules of deviceElectricMotorcycle Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| runState | runState | 1 | See clause 5.3.1.75.  This module class is used to control and monitor the state of an electric motorcycle. |
| battery | battery | 1 | See clause 5.3.1.10 |
| geoLocation | geoLocation | 1 | See clause 5.3.1.41 |
| clock | clock | 0..1 | See clause 5.3.1.18 |
| faultDetection | faultDetection | 0..1 | See clause 5.3.1.34 |
| lock | lock | 0..1 | See clause 5.3.1.52 |

### Agriculture Domain

#### deviceCowActivityMonitor

A cow activity monitor device is a battery-powered wireless metering device that is used to measure the activity data (e.g. step count) of a cow in the dairy farming industry. The collected data can be used to analyze and predict the estrus of the cow for better mating and milk production.

Table 5.5.7.1-1: Modules of deviceCowActivityMonitor Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| cowActivityMonitor | cowActivityMonitor | 1 | See clause5.3.7.1. |
| battery | battery | 0..1 | See clause 5.3.1.10. |
| connectivity | connectivity | 0..1 | See clause 5.3.1.25. |
| periodicalReportConfig | periodicalReportConfig | 0..1 | See clause 5.3.1.63 |

### Railway Domain

#### deviceHandheldPTTTerminal

A handheld PTT (Push-to-Talk) terminal is a device of the Railway Domain. The terminal is usually used between the railway workers including train driver, crew and rail-side worker to share their work status via voice communication.

Table 5.5.8.1-1: Modules of deviceHandheldPTTTerminal Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| audioVolume | audioVolume | 1 | See clause 5.3.1.8 |
| battery | battery | 1 | See clause 5.3.1.10. |
| clock | clock | 0..1 | See clause 5.3.1.18. |
| energyConsumption | energyConsumption | 0..1 | See clause 5.3.1.32. |
| keypad | keypad | 1 | See clause 5.3.1.49. |
| operationMode | operationMode | 0..1 | See clause 5.3.1.57. |
| phoneCall | phoneCall | 1 | See clause 5.3.1.64. |
| PTTButton | pushButton | 1 | See clause 5.3.1.69. |
| runState | runState | 0..1 | See clause 5.3.1.75. |
| signalStrength | signalStrength | 0..1 | See clause 5.3.1.78. |

#### deviceTrainborneTerminal

A handheld PTT (Push-to-Talk) terminal is a device of the Railway Domain. The terminal is usually used between the railway workers including train driver, crew and rail-side worker to share their work status via voice communication.

Table 5.5.8.2-1: Modules of deviceTrainborneTerminal Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| audioVolume | audioVolume | 1 | See clause 5.3.1.8 |
| clock | clock | 0..1 | See clause 5.3.1.18. |
| energyConsumption | energyConsumption | 0..1 | See clause 5.3.1.32. |
| keypad | keypad | 1 | See clause 5.3.1.49. |
| operationMode | operationMode | 0..1 | See clause 5.3.1.57. |
| phoneCall | phoneCall | 1 | See clause 5.3.1.64. |
| PTTButton | pushButton | 1 | See clause 5.3.1.69. |
| runState | runState | 0..1 | See clause 5.3.1.75. |
| signalStrength | signalStrength | 0..1 | See clause 5.3.1.78. |
| baliseTransmission | baliseTransmissionModule | 0..1 | See clause 5.3.8.1 |
| connectivity | connectivity | 0..1 | See clasue 5.3.1.25 |

#### deviceCardRechargingMachine

A card recharging machine is a device of the Railway Domain. The machine provides recharging service for pre-paid card for transportation. Railway users simply recharging their cards and use it as a payment method for transportation fare.

Table 5.5.8.3-1: Modules of deviceCardRechargingMachine Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| clock | clock | 1 | See clause 5.3.1.18. |
| touchScreen | touchScreen | 1 | See clause 5.3.1.106. |
| keypad | keypad | 0..1 | See clause 5.3.1.49. |
| emergencyButton | pushButton | 1 | See clause 5.3.1.69. |
| audioVolume | audioVolume | 0..1 | See clause 5.3.1.8. |
| prePaidCardReader | prePaidCardReader | 1 | See clause 5.3.1.107. |
| billDeposit | billDeposit | 1 | See clause 5.3.1.108. |
| billWithdrawal | billWithdrawal | 1 | See clause 5.3.1.109. |
| coinDeposit | coinDeposit | 1 | See clause 5.3.1.110. |
| cashDispensor | cashDispenser | 1 | See clause 5.3.1.111. |
| cardScanner | cardScanner | 0..1 | See clause 5.3.1.112. |
| connectivity | connectivity | 1 | See clause 5.3.1.25. |
| machineState | runState | 1 | See clause 5.3.1.75. |
| lock | lock | 1 | See clause 5.3.1.52. |

#### deviceSmartGate

A smart gate is a device of the Railway Domain. The gate provides passenger service for checking ticket and control the gate usage. A railway users simply use the gate with their ticket (e.g. passenger card) which have pre-paid card function.

Table 5.5.8.4-1: Modules of deviceSmartGate Device model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| clock | clock | 1 | See clause 5.3.1.18. |
| touchScreen | touchScreen | 0..1 | See clause 5.3.1.106. |
| emergencyButton | pushButton | 0..1 | See clause 5.3.1.69. |
| audioVolume | audioVolume | 0..1 | See clause 5.3.1.8. |
| ticketReader | prePaidCardReader | 1 | See clause 5.3.1.107. |
| crossingSensor | traceSensor | 1 | See clause 5.3.1.113. |
| connectivity | connectivity | 1 | See clause 5.3.1.25. |
| gateState | runState | 1 | See clause 5.3.1.75. |
| directionPanel | directionPanel | 0..1 | See clause 5.3.1.114. |
| crossingIndicatorColour | colour | 1 | See clause 5.3.1.23. |
| crossingIndicatorColourSaturation | colourSaturation | 1 | See clause 5.3.1.24. |
| crossingIndicatorColourBrightness | brightness | 1 | See clause 5.3.1.17. |
| crossingBarrier | crossingBarrier | 1 | See clause 5.3.1.115. |

#### deviceSmartScreenDoor

A smart screen door is a device of the Railway Domain. The screen doors are pairs of sliding doors in a platform with synchronization of doors of a train that is staying in the platform. When the train doors are open, the smart screen doors are open in simultaneously, and vice versa. The screen doors are used in the metro platform mostly.

Table 5.5.8.5-1: Modules of deviceSmartScreenDoor Device Model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| binarySwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| screenDoor | doorStatus | 1..N | See clause 5.3.1.30. |
| releaseSwitch | binarySwitch | 1 | See clause 5.3.1.12. |
| emergencyButton | pushButton | 1 | See clause 5.3.1.69. |
| audioVolume | audioVolume | 0..1 | See clause 5.3.1.8. |
| crossingSensor | traceSensor | 1 | See clause 5.3.1.113. |
| connectivity | connectivity | 1 | See clause 5.3.1.25. |
| gateState | runState | 1 | See clause 5.3.1.75. |
| directionPanel | directionPanel | 0..1 | See clause 5.3.1.114. |
| crossingIndicatorColour | colour | 0..1 | See clause 5.3.1.23. |
| crossingIndicatorColourSaturation | colourSaturation | 0..1 | See clause 5.3.1.24. |
| crossingIndicatorColourBrightness | brightness | 0..1 | See clause 5.3.1.17. |

## Enumeration type definitions

All enumeration types are defined in the same domain, Horizontal Domain, prefix ‘hd’.

### hd:enum3DprinterTechnology

Used for the “printType” data point of the “threeDprinter” ModuleClass.

Table 5.6.1‑1: Interpretation of hd:enum3DprinterTechnology

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | Fused Filament Fabrication | FFF |
| 2 | Fused Deposition Modeling | FDM |
| 3 | Digital Light Processing | DLP |
| 4 | Powder Bed & inkjet head 3D Printing | PBP |
| 5 | Photopolymer Jetting Technology | PolyJet |
| 6 | Laminated Object Manufacturing | LOM |
| 7 | Stereolithography Apparatus | SLA |
| 8 | Selective Laser Sintering | SLS |
| NOTE: See clause 5.3.1.1 "threeDprinter". | | |

### hd:enumAdfState

Used for the “currentAdfState” and “adfStates” data points of the “autoDocumentFeeder” ModuleClass.

Table 5.6.2‑1: Interpretation of hd:enumAdfState

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | processing |  |
| 2 | empty |  |
| 3 | jam |  |
| 4 | loaded |  |
| 5 | mispick | The product did not pick up the paper in the document feeder. |
| 6 | hatchOpen | The product hatch is open. |
| 7 | duplexPageTooShort |  |
| 8 | duplexPageTooLong |  |
| 9 | multipickDetected |  |
| 10 | inputTrayFailed |  |
| 11 | inputTrayOverloaded |  |
| NOTE: See clause 5.3.1.9 "autoDocumentFeeder". Negative values are reserved for vendor specific modes. | | |

### hd:enumAirConJobMode

Used for the “currentJobMode” and “jobModes” data point of the “airConJobMode” ModuleClass.

Table 5.6.3‑1: Interpretation of hd:enumAirConJobMode

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | cool | This value is for deviceAirConditioner and indicates cool mode. |
| 2 | airDry | This value is for deviceAirConditioner and indicates air dry mode. |
| 3 | fan | This value is for deviceAirConditioner and indicates fan mode. |
| 4 | AI | This value is for deviceAirConditioner and indicates artificial intelligence mode. |
| 5 | heat | This value is for deviceAirConditioner and indicates heat mode. |
| 6 | airClean | This value is for deviceAirConditioner and indicates air clean mode. |
| 7 | ACO | This value is for deviceAirConditioner and indicates Auto Change Over mode. |
| 8 | aroma | This value is for deviceAirConditioner and indicates aroma mode. |
| NOTE: See clause 5.3.1.3 "airConJobMode". Negative values are reserved for vendor specific modes. | | |

### hd:enumAirPurifierJobMode

Used for the “currentJobMode” and “jobModes” data points of the “airPurifierJobMode” ModuleClass.

Table 5.6.4‑1: Interpretation of hd:enumAirPurifierJobMode

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | normalClean | This indicates the normal mode that operates the basic function. |
| 2 | sleep | This indicates the sleep mode that turns the operating function off at the time set by a timer. |
| 3 | silent | This indicates the silent mode that generates low noise |
| 4 | wet | This indicates the wet mode that passes the air that’s already filtered through water filter once again to provide the humidification effect |
| 5 | circulate | This indicates the circulate mode that circulates the purified air by rotating the fan on top of the air purifier |
| 6 | dual | This indicates the dual mode that operates both the upper and lower parts of the air purifier |
| 7 | auto | This indicates the auto mode that first measures the pollution level (e.g., good, normal, bad, very bad) and then, triggers appropriate modes based on the measured level. In case of bad and very bad condition, the rotating fan on the upper side starts its operation |
| NOTE: See clause 5.3.1.5 "airPurifierJobMode". Negative values are reserved for vendor specific modes. | | |

### hd:enumAlertColourCode

Used for the “light” data point of the “alarmSpeaker” ModuleClass.

Table 5.6.5‑1: Interpretation of hd:enumAlertColourCode

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | red | This colour indicates the alarm status. |
| 2 | green | This colour indicates the alarm has been cleared. |
| NOTE: See clause 5.3.1.7 "alarmSpeaker". | | |

### hd:enumCallState

Used for the “callState” data point in the “phoneCall” ModuleClass.

Table 5.6.6‑1: Interpretation of hd:enumCallState

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | hangup |  |
| 2 | calling |  |
| 3 | ringing |  |
| 4 | busy |  |
| 5 | answered |  |
| 6 | noline |  |
| 7 | voicemail |  |
| 8 | redirected |  |
| NOTE: See clause 5.3.1.64 “phoneCall” | | |

### hd:enumClothesDryerJobMode

Used for the “currentJobMode” and “jobModes” data points of the “clothesDryerJobMode” ModuleClass.

Table 5.6.7‑1: Interpretation of hd:enumClothesDryerJobMode

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | normal | Normal cycle. |
| 2 | quickDry | About half the length of a normal cycle, this setting uses high heat to dry a few items. |
| 3 | permanentPress | Slow drying with low heat helps wringkle-free garments live up to their name and keeps the hard creases out of thins you typically iron. |
| 4 | heavyDuty | Tumbling for an extended period with high heat for sturdy items (towels, sweats, jeans). |
| 5 | delicates | A short, low-heat cycle for delicates and other items such as spandex workout gear, which loses its stretch when too much heat is used. |
| 6 | airDry | A cool-air setting for items that can’t take any heat, such as plastic tablecloths and rubber-backed rugs. |
| 7 | extendedTumble | Periodically tumbles clothes without heats for a preset amount of time after they’re dry to prevent wrinkles. |
| NOTE: See clause 5.3.1.19 "clothesDryerJobMode". Negative values are reserved for vendor specific modes. | | |

### hd:enumClothesWasherDryerJobMode

Used for “currentJobModes” and “jobModes” data points of “clothesWasherDryerJobMode” ModuleClass. Manufacturers can define their own courses by setting this value to negative values.

Table 5.6.8‑1: Interpretation of hd:enumClothesWasherJobMode

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | normal |  |
| 2 | quick |  |
| 3 | auto |  |
| 4 | delicates |  |
| 5 | heavy duty |  |
| NOTE: See clause 5.3.1.21 “clothesWasherDryerJobMode”. Negative values are reserved for vendor specific modes. | | |

### hd:enumClothesWasherJobMode

Used for the “currentJobModes” and “jobModes ” data points of the “clothesWasherJobMode” ModuleClass. Washing options such as water temperature and spin speed are decided to pre-set values upon selected washing course. Manufacturers can define their own courses by setting this value to negative values.

Table 5.6.9‑1: Interpretation of hd:enumClothesWasherJobMode

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | normal |  |
| 2 | smallLoad |  |
| 3 | delicate |  |
| 4 | comforter |  |
| 5 | expressWash |  |
| 6 | cleanWash |  |
| 7 | kidsWear |  |
| 8 | workoutWears |  |
| NOTE: See clause 5.3.1.20 “clothesWashingJobMode”. Negative values are reserved for vendor specific modes. | | |

### hd:enumCookerHoodJobMode

Used for the “currentJobMode” and “jobModes” DataPoints of the “cookerHoodJobMode” ModuleClass.

Table 5.6.10‑1: Interpretation of hd:enumCookerHoodJobMode

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | Always-on | This value indicates the always-on mode which keeps running the fan for ventilation. |
| 2 | Intensive | This value indicates the intensive mode used when a large volume of cooking fume is being produced. |
| 3 | Sensor | This value indicates the sensor mode which changes fan speed depend on the volume and heat of cooking fume. |
| NOTE: See clause 5.3.1.26 "cookerHoodJobMode". Negative values are reserved for vendor specific modes. | | |

### hd:enumDehumidifierJobMode

Used for “currentJobMode” and “jobModes” data points of the “dehumidifierJobMode” ModuleClass.

Table 5.6.11‑1: Interpretation of hd:enumDehumidifierJobMode

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | smart | This value indicates the smart mode that first gets the target humidity level from user input, next detects the currect relative humidity, then automatically change the dehumidity level to keep the target humidity level. |
| 2 | fast | This value indicates the fast mode that speeds the operating level up to quickly dehumidify when the humidity level is so high. It is a kind of turbo mode. |
| 3 | silent | This value indicates the silent mode that can be used when an user sleeps. It reduces the noise. |
| 4 | focus | This value indicates the focus mode that dehumidifies focusing on a particular part. |
| 5 | clothes | This value indicates the clothes mode that dehumidifies adjusting the wind direction vertically. It is normally used to dehumidify clothes. |
| NOTE: See clause 5.3.1.28 "dehumidifierJobMode". Negative values are reserved for vendor specific modes. | | |

### hd:enumDishWasherJobMode

Used for the “currentJobMode” and “jobModes” DataPoints of the “dishWasherJobMode” ModuleClass.

Table 5.6.12‑1: Interpretation of hd:enumDishWasherJobMode

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | Normal wash |  |
| 2 | Intensive wash |  |
| 3 | Quick wash |  |
| 4 | Sensor wash |  |
| 5 | Eco wash |  |
| 6 | Quiet wash |  |
| 7 | Maintenance wash |  |
| NOTE: See clause 5.3.1.29 "dishWasherJobMode". Negative values are reserved for vendor specific modes. | | |

### hd:enumDisplayOrder

Used for the “displayOrder” data point of the “galleryMode” ModuleClass.

Table 5.6.13-1: Interpretation of hd:enumDisplayOrder

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | fixed |  |
| 2 | sequence |  |
| 3 | loop |  |
| 4 | random |  |

### hd:enumDisplayOrientation

Used for the “displayOrientation” data point of the “galleryMode” ModuleClass.

Table 5.6.14-1: Interpretation of hd:enumDisplayOrientation

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | landscape |  |
| 2 | portrait |  |

### hd:enumDoorState

Used for the “doorState” DataPoint of “doorStatus” ModuleClass.

Table 5.6.15‑1: Interpretation of hd:enumDoorState

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | closed | This indicates that door is closed. |
| 2 | open | This indicates that the door is open. |
| 3 | opening | This indicates that the door is opening. |
| 4 | closing | This indicates that the door is closing. |
| 5 | stopped | This indicates that the door is in stationary state. |
| NOTE: See clause 5.3.1.29 "doorStatus". | | |

### hd:enumFoamStrength

Used for data points indicating the strength of a foam, for example, foaming milk from a coffee machine.

Table 5.6.16‑1: Interpretation of hd:enumFoamStrength

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | zero |  |
| 2 | low |  |
| 3 | medium |  |
| 4 | high |  |
| 5 | maximum |  |
| NOTE: See clause 5.3.1.36 “foaming” | | |

### hd:enumGeneralLevel

Used for the “soilLevel” data point of the “washingCourseOption” ModuleClass.

Table 5.6.17‑1: Interpretation of hd:enumGeneralLevel

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | light |  |
| 2 | normal |  |
| 3 | heavy |  |
| NOTE: See clause 5.3.1.22 clothesWasherJobModeOption | | |

### hd:enumGeneralSpeed

Used for the “spinSpeed” data point of the “washingCourseOption” ModuleClass.

Table 5.6.18‑1: Interpretation of hd:enumGeneralSpeed

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | low |  |
| 2 | medium |  |
| 3 | high |  |
| 4 | extraHigh |  |
| NOTE: See clause 5.3.1.22 clothesWasherJobModeOption | | |

### hd:enumGeneralTemperature

Used for the “washTemp” data point of the “washingCourseOption” ModuleClass.

Table 5.6.19‑1: Interpretation of hd:enumGeneralTemperature

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | cold | The actual temperature is defined by the manufacturer. |
| 2 | warm |  |
| 3 | hot |  |
| NOTE: See clause 5.3.1.22 clothesWasherJobModeOption | | |

### hd:enumGrainsLevel

Used for the “grainsRemaining” data point of the “grinder” ModuleClss. This type specifies a level for supplies that have a grain-aspect, for example.the level of remaining coffee beans in the grinder part of a coffee machine, or the desired level of coffee beans in this machine.

The values for the level of a liquid is covered by “hd:enumLiquidLevel” (see clause 5.6.20).

Table 5.6.20‑1: Interpretation of hd:enumGrainsLevel

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | zero |  |
| 2 | low |  |
| 3 | medium |  |
| 4 | high |  |
| 5 | maximum |  |
| NOTE: See clause 5.3.1.43 "grinder". | | |

### hd:enumGrindCoarseness

Used for the coarseness data points of the “grinder” ModuleClass. This type specifies the level of coarseness of a solid after grinding, for example grinded coffee beans.

Table 5.6.21‑1: Interpretation of hd:enumGrindCoarseness

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | ultrafine |  |
| 2 | fine |  |
| 3 | medium |  |
| 4 | coarse |  |
| 5 | coarsest |  |
| NOTE: See clause 5.3.1.43 “grinder” | | |

### hd:enumHorizontalDirection

Used for the “horizontalDirection” and “supportedHorizontalDirection” of the “airflow” ModuleClass, indicating horizontal directions.

Table 5.6.22‑1: Interpretation of hd:enumHorizontalDirection

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | auto |  |
| 2 | center |  |
| 3 | left |  |
| 4 | right |  |
| NOTE: See clause 5.3.1.4 "airFlow". | | |

### hd:enumJobStates

Used for the “currentJobState” and “jobStates” data points of the “runState” ModuleClass.

Table 5.6.23‑1: Interpretation of hd:enumJobState

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | aborted |  |
| 2 | cancelled |  |
| 3 | completed |  |
| 4 | paused |  |
| 5 | pending |  |
| 6 | processing |  |
| NOTE: See clause 5.3.1.75 "runstate". | | |

### hd:enumLiquidLevel

Used for the “liquidLevel“ and “liquidRemaining” data points in the respective “liquidLevel” and “liquidRemaining” ModuleClasses.

Table 5.6.24‑1: Interpretation of hd:LiquidLevel

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | zero |  |
| 2 | low |  |
| 3 | medium |  |
| 4 | high |  |
| 5 | maximum |  |
| NOTE: See clause 5.3.1.50 "liquidLevel" and clause 5.3.1.51 “liquidRemaining”. | | |

### hd:enumMachineState

Used for the “currentMachineState” and “machineStates” data points of the “runState” ModuleClass.

Table 5.6.25‑1: Interpretation of hd:enumMachineState

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | idle | Machine is ready to operate |
| 2 | preActive | Machine is operating its pre-functions (ex. pre-heat) |
| 3 | active | Machine is operating its functions |
| 4 | reserved | Reservation is made by user |
| 5 | stopped | Operation is stopped/aborted by some other reasons |
| 6 | error | Error has occurred |
| 7 | diagnostic | Machine reports diagnostic information to the server |
| 8 | test | Particular functions run for test |
| 9 | maintenance | Machine is needed to maintain |
| 10 | clear | The result is not removed yet |
| 11 | charging | Machine is being charged |
| NOTE: See clause 5.3.1.75 "runState". | | |

### hd:enumOzoneStatus

Used for the “ozoneStatus” property of the “ozoneMeter” ModuleClass.

Table 5.6.26‑1: Interpretation of hd:enumOzoneStatus

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | Good | For example, 0 to 0.030 ppm. |
| 2 | Normal | For example, 0.031 to 0.090 ppm. |
| 3 | Bad | For example, 0.091 to 0.150ppm. |
| 4 | Very bad | For example, 0.151 ppm or above. |
| NOTE: See clause 5.3.1.60 "ozoneMeter".  The examples in the notes are references from Korean Environmental Standard[i.7]. | | |

### hd:enumPlayerMode

Used for the “currentMode” and “supportedModes” data points in the “playerControl” ModuleClass.

Table 5.6.27‑1: Interpretation of hd:enumPlayerMode

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | stop |  |
| 2 | play |  |
| 3 | pause |  |
| 4 | resume |  |
| 5 | record |  |
| 6 | rewind |  |
| 7 | fast-rewind |  |
| 8 | forward |  |
| 9 | fast-forward |  |
| 10 | searchPrevious |  |
| 11 | searchNext |  |
| NOTE: See clause 5.3.1.65 “playerControl” | | |

### hd:enumRobotCleanerJobMode

Used for the “currentJobMode” and “jobModes” data points of the “robotCleanerJobMode” ModuleClass.

Table 5.6.28‑1: Interpretation of hd:enumRobotCleanerJobMode

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | zigzag | The machine moves forward by going at an angle first to one side then to the other |
| 2 | sectorBase | The machine first cleans a specific sector (fo example, 1x1m), then moves to another sector. |
| 3 | spot | The machine cleans a targeted area of about specific spot |
| NOTE: See clause 5.3.1.74 "robotCleanerJobMode". Negative values are reserved for vendor specific modes. | | |

### hd:enumSecurityMode

Used for the “currentSecurityMode” and “securityModes” data points of the “securityMode” ModuleClass.

Table 5.6.29‑1: Interpretation of hd:enumSecurityMode

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | active | Unit is active |
| 2 | armedAway | Unit is armed for away |
| 3 | armedInstant | Unit is armed instantly |
| 4 | armedMaximum | Unit is armed at maximum level |
| 5 | armedNightStay | Unit is armed in night stay |
| 6 | armedStay | Unit is armed in stay mode |
| NOTE: See clause 5.3.1.76 “securityMode” | | |

### hd:enumSpinLevelStrength

Used for the “spinLevelStrength” data points of the “spinLevel” ModuleClass, and used for the “spinSpeed” data points of the “blender” ModuleClass, indicating the strength of a spinLevel.

Table 5.6.30‑1: Interpretation of hd:enumSpinLevelStrength

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | zero |  |
| 2 | sensitive |  |
| 3 | weak |  |
| 4 | medium |  |
| 5 | strong |  |
| 6 | maximum |  |
| NOTE: See clause 5.3.1.84 "spinLevel" and clause 5.3.1.118 ”blender”. | | |

### hd:enumSteamClosetJobMode

Used for “currentJobMode” and “jobModes” data points of the “steamClosetJobMode” ModuleClass.

Table 5.6.31‑1: Interpretation of hd:enumSteamClosetJobMode

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | reduceOdor | Using pure water, the machine help remove the smells on clothes. |
| 2 | steamWrinkle | The machine steams away wrinkles and also creates pant creases, as well as keep them crisp. |
| 3 | helpClean | Using pure water without chemical additives, the machine sanitizes fabrics and items that are difficult to wash. |
| 4 | gentleDry | The machine dries fragile garments without worrying about shrinkage or damage. |
| NOTE: See clause 5.3.1.85 "steamClosetJobMode". Negative values are reserved for vendor specific modes. | | |

### hd:enumSupportedMediaSources

Used for the “supportedMediaSources” data point of the “mediaSelect” ModuleClass.

Table 5.6.32‑1: Interpretation of hd:enumSupportedMediaSources

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | tuner |  |
| 2 | component |  |
| 3 | composite |  |
| 4 | svideo |  |
| 5 | rgb |  |
| 6 | dvi |  |
| 7 | hdmi |  |
| 8 | displayPort |  |
| 9 | scart |  |
| 10 | externalStorage |  |
| 11 | network |  |
| NOTE: See clause 5.3.1.53"mediaSelect". Negative values are reserved for vender specific sources. | | |

### hd:enumTasteStrength

Used for the “strength” data point of the “brewing” ModuleClass, indicating strength of a drink taste, for example coffee strength.

Table 5.6.33‑1: Interpretation of hd:enumTasteStrength

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | zero |  |
| 2 | sensitive |  |
| 3 | medium |  |
| 4 | strong |  |
| 5 | maximum |  |
| NOTE: See clause 5.3.1.16 “brewing” | | |

### hd:enumTone

Used for the “tone” data point of the “alarmSpeaker” ModuleClass.

Table 5.6.34‑1: Interpretation of hd:enumTone

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | fire |  |
| 2 | theft |  |
| 3 | emergency |  |
| 4 | doorbell |  |
| 5 | deviceFail |  |
| NOTE: See clause 5.3.1.7 "alarmSpeaker". | | |

### hd:enumUvStatus

Used for the “uvStatus” data point of the “uvSensor” ModuleClass.

Table 5.6.35‑1: Interpretation of hd:enumUvStatus

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | Good |  |
| 2 | Normal |  |
| 3 | Bad |  |
| 4 | Very Bad |  |
| 5 | Danger |  |
| NOTE: See clause 5.3.1.92 "uvSensor". | | |

### hd:enumVerticalDirection

Used for the “verticalDirection” and “supportedVerticalDirection” data points of the “airFlow" ModuleClass, indicating vertical direction.

Table 5.6.36‑1: Interpretation of hd:enumVerticalDirection

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | auto |  |
| 2 | center |  |
| 3 | up |  |
| 4 | down |  |
| NOTE: See clause 5.3.1.4 "airFlow". | | |

### hd:enumWaterFlowStrength

Used for the “waterLevelStrength” data point of the “waterFlow” ModuleClass, indicating the strength of a waterflow.

Table 5.6.37‑1: Interpretation of hd:enumWaterFlowStrength

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | zero |  |
| 2 | sensitive |  |
| 3 | weak |  |
| 4 | medium |  |
| 5 | strong |  |
| 6 | maximum |  |
| NOTE: See clause 5.3.1.93 "waterFlow". | | |

### hd:enumBaliseSystemIndicator

Used for the “baliseTransmissionModule” ModuleClass.

Table 5.6.38‑1: Interpretation of hd:enumBaliseSystemIndicator

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | ATC | Korea [i.9] |
| 2 | ATP | Korea [i.9] |
| 3 | CTCS-3 | China [i.10] |
| NOTE: The Note shows countries which are using a balise system which is indicated on left-hand side. | | |

### hd:enumWeight

Used for the “unit” data point related to all ModuleClass which contains weight, indicating the units of the weight..

Table 5.6.39‑1: Interpretation of hd:enumWeight

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | kg | kilogram |
| 2 | lb | pound |
| 3 | oz | ounce |
| NOTE: See clause 5.3.1.99 "weight". | | |

### hd:enumTemperatureUnit

Used for the “unit” data point related to “temperature” and “temperatureAlarm” ModuleClass which contains temperature, indicating the units of the temperature.

Table 5.6.40‑1: Interpretation of hd:enumTemperatureUnit

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | C | Celsius |
| 2 | F | Fahrenheit |
| 3 | K | Kelvin |
| NOTE: See clause 5.3.1.87 "temperature" and clause 5.3.1.88 “temperatureAlarm”. | | |

### hd:enumWaterFilterType

Used for the “filterType” data point of the “waterFilterType” ModuleClass.

Table 5.6.41‑1: Interpretation of hd:enumWaterFilterType

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | RO | This value indicates the Revers Osmosis type water filter. |
| 2 | UV | This value indicates the Ultraviolet type water filter. |
| 3 | UF | This value indicates the UltraFiltration type water filter. |
| 4 | AC | This value indicates the Activate Carbon type water filter. |
| 5 | SF | This value indicates the Sediment type water filter. |
| NOTE: See clause 5.3.1.105 "waterFilterType". | | |

### hd:enumDataModelType

Used for the “dataModelType” DataPoint of the “dmAgent” ModuleClass.

Table 5.6.42-1 Interpretation of hd:enumDataModelType

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | unknown | To be used for a proprietary/unknown protocol |
| 2 | OMA\_DM\_1.3 |  |
| 3 | OMA\_DM\_2.0 |  |
| 4 | OMA\_LwM2M |  |
| 5 | BBF\_TR-181\_CWMP | Version of TR-181 for TR-069 |
| 6 | BBF\_TR-181\_USP | Version of TR-181 for USP |
| 7 | oneM2M | For native oneM2M devices |

### hd:enumDmAgentState

Used for the “state” DataPoint of the “dmAgent” ModuleClass.

Table 5.6.43-1 Interpretation of hd:enumDmAgentState

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | ready | The device is ready for DM operations |
| 2 | error | The device is known to be in an error state |
| 3 | sleeping | The device is known to be in sleeping mode |
| 4 | unreachable | The device is not accessible |

### hd:enumFirmwareState

Used for the “state” DataPoint of the “dmFirmware” ModuleClass.

Table 5.6.44-1 Interpretation of hd:enumFirmwareState

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | Active | The firmware is currently active |
| 2 | Ready | The firmware is ready for installation/activation |
| 3 | Downloading | The firmware is being downloaded |
| 4 | Installing | The firmware in being installed |
| 5 | Failure | The firmware installation/download has failed |
| 6 | Archival | The firmware is an archival record that cannot be reactivated. |

### hd:enumPowerState

Used for the “powerStatus” DataPoint of the “dmAgent” ModuleClass.

Table 5.6.45-1 Interpretation of hd:enumPowerState

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | normal |  |
| 2 | charging |  |
| 3 | chargingComplete |  |
| 4 | degraded |  |
| 5 | low |  |
| 6 | critical |  |
| 7 | notInstalled |  |

### hd:enumRebootType

Used for the “rebootType” argument of the “reboot” action of the “dmAgent” ModuleClass.

Table 5.6.46-1 Interpretation of hd:enumRebootType

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | reboot |  |
| 2 | rebootWhenReady | Reboot needed when the device is available for it |
| 3 | factoryReset | “hard” reset |
| 4 | softReset |  |

### hd:enumSoftwareState

Used for the “state” data point of the “dmSoftware” ModuleClass.

Table 5.6.47-1 Interpretation of hd:enumSoftwareState

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | Inactive |  |
| 2 | Activating |  |
| 3 | Active |  |
| 4 | Deactivating |  |

### hd:enumPackageState

Used for the “state” data point of the “dmPackage” ModuleClass.

Table 5.6.48-1 Interpretation of hd:enumPackageState

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | NotInstalled |  |
| 2 | Downloaded |  |
| 3 | Installed |  |
| 4 | Downloading |  |
| 5 | Installing |  |
| 6 | Uninstalling |  |

### hd:enumPackageType

Used for the “type” data point of the “dmPackage” ModuleClass.

Table 5.6.49-1 Interpretation of hd:enumPackageType

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | SoftwareModule | Software module image (executable) |
| 2 | SoftwareLibrary | Software library file |
| 3 | WebContent | Web document |
| 4 | ConfigFile | Configuration file |
| 5 | VendorFile | Vendor-specific document |
| 6 | Undefined |  |

### hd:enumBatteryMaterial

Used for the “batteryMaterial” DataPoint of the “battery” ModuleClass.

Table 5.6.50-1 Interpretation of hd:enumBatteryMaterial

|  |  |  |
| --- | --- | --- |
| **Value** | **Interpretation** | **Note** |
| 1 | Alkaline\_battery | Primary cells or non-rechargeables |
| 2 | Lithium\_battery | Primary cells or non-rechargeables |
| 3 | Magnesium\_battery | Primary cells or non-rechargeables |
| 4 | Mercury\_battery | Primary cells or non-rechargeables |
| 5 | Nickel\_oxyhydroxide\_battery | Primary cells or non-rechargeables |
| 6 | Silver\_oxide\_battery | Primary cells or non-rechargeables |
| 7 | Zinc\_air | Primary cells or non-rechargeables |
| 8 | Lead\_acid\_battery | Secondary cells or rechargeables |
| 9 | Lithium\_ion\_battery | Li-ion, Secondary cells or rechargeables |
| 10 | Lithium\_ion\_polymer\_battery | LiPo, Secondary cells or rechargeables |
| 11 | Nickel\_cadmium\_battery | Ni-Cd , Secondary cells or rechargeables |
| 12 | Nickel\_iron\_battery | Secondary cells or rechargeables |
| 13 | Nickel\_metal\_hydride\_battery | NiMH, Secondary cells or rechargeables |
| 14 | Nickel\_zinc\_battery | Secondary cells or rechargeables |
| 15 | Rechargeable\_alkaline\_battery | Secondary cells or rechargeables |

### hd:enumBatteryShape

Used for the “batteryShape” DataPoint of the “battery” ModuleClass.

Table 5.6.51-1 Interpretation of hd:enumBatteryShape

|  |  |  |
| --- | --- | --- |
| **Value** | **Interpretation** | **Note** |
| 1 | AA | Cylinder-type AA battery |
| 2 | AAA | Cylinder-type AAA battery |
| 3 | AAAA | Cylinder-type AAAA battery |
| 4 | C | Cylinder-type C battery |
| 5 | D | Cylinder-type D battery |
| 6 | N | Cylinder-type N battery |
| e | A23 | Cylinder-type A23 battery |
| 8 | Coin\_cell\_4 | Coin-cell type 4.8 mm diameter battery |
| 9 | Coin\_cell\_5 | Coin-cell type 5.8 mm diameter battery |
| 10 | Coin\_cell\_6 | Coin-cell type 6.8 mm diameter battery |
| 11 | Coin\_cell\_7 | Coin-cell type 7.9 mm diameter battery |
| 12 | Coin\_cell\_9 | Coin-cell type 9.5 mm diameter battery |
| 13 | Coin\_cell\_10 | Coin-cell type 10.0 mm diameter battery |
| 14 | Coin\_cell\_11 | Coin-cell type 11.6 mm diameter battery |
| 15 | Coin\_cell\_12 | Coin-cell type 12.5 mm diameter battery |
| 16 | Coin\_cell\_16 | Coin-cell type 16.0 mm diameter battery |
| 17 | Coin\_cell\_20 | Coin-cell type 20.0 mm diameter battery |
| 18 | Coin\_cell\_23 | Coin-cell type 23.0 mm diameter battery |
| 19 | Coin\_cell\_24 | Coin-cell type 24.5 mm diameter battery |
| 20 | Coin\_cell\_44 | Coin-cell type 5.4 mm diameter battery |
| 21 | Box\_9V | Box type 9V battery |
| 22 | Silver\_Flat\_Pack | Flat Box Pack type |
| 23 | Car\_battery | Box type 6-cell lead car battery |
| 24 | Custom\_made | Custom-made by manutacturer |

### hd:enum3DDisplayType

Used for the “threeDDisplayType” DataPoint of the “threeDDisplay” ModuleClass.

Table 5.6.52-1 Interpretation of hd:enum3DDisplayType

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | Stereoscopic\_Display | Use “binocular disparity” method to implement 3D display. The technology uses 3D glasses to make the binocular disparity. |
| 2 | Light\_Field\_Display | The technology build barriers or lenticular lens on a RGB panel to make binocular disparity effect. It does not need 3D glasses. |
| 3 | Volumetric\_Display | The technology uses the interference ray as the light source. The 3D image is formed as a set of pixels that the pixels are generated as a bright point in the position of constructive interference is made. This technology does not need 3D glasses. |
| 4 | ETC |  |

### hd:enum3DScannerType

Used for the “3DScannerType” DataPoint of the “3DScanner” ModuleClass.

Table 5.6.53-1 Interpretation of hd:enum3DScannerType

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | TOF | Use “Time Of Flight” method to scan 3D object. The technology calculates time gap between shooting and return of the reflected laser light. |
| 2 | Phase\_Shift | The technology uses “Phase shift waveform analysis”. It analysises the distance gap between two reflected laser beam phase which are shot from the scanner. |
| 3 | Waveform | The technology uses “Triangulation method”. Based on the triangulation method, it uses pointbeam or TOF method. |
| 4 | MPT | The technology uses “Miniaturized Projection Technique” to scan. It projects specific pattern of White light, indicates the size and depth by analysis of the reflected pattern on the object. |
| 5 | ETC | The other technology is used to scan an object. |

### hd:enumShoesWasherJobMode

Used for the “currentJobMode” and “jobModes” DataPoints of the “shoedWasherJobMode” ModuleClass.

Table 5.6.54‑1: Interpretation of hd:enumShoesWasherJobMode

|  |  |  |
| --- | --- | --- |
| Value | Interpretation | Note |
| 1 | normalWash |  |
| 2 | intensiveWash |  |
| 3 | quickWash |  |
| 4 | maintenanceWash |  |
| 5 | antibacterialWash |  |
| NOTE: See clause 5.3.1.119 "shoesWasherJobMode". Negative values are reserved for vendor specific modes. | | |

## Universal and Common Properties for Device models

Universal and common properties are defined either

* As specialized custom attributes of a [dmDeviceInfo] <flexContainer> specialization, defined in 5.8.4, when the <node> resource targeted by the *nodeLink* attribute of the Device model contains a [flexNode] child,
* Or as specialized object attributes of the [deviceInfo] <mgmtObj> specialization, defined in Annex D.8 of TS-0001[3], child of this <node> resource, otherwise.

Some properties are mandatory for all device models and called "Universal Properties", since they are universally seen in typical device types and carry necessary information to identify each device instance. Others are optional for all device models and called "Common Properties", since they are commonly used in many device types but not always.

Universal and common properties are applicable to all device models. They are not repeated in the property table of each device model in clause 5.5, where only device specific properties shall be specified.

NOTE: The instantiated values of the universal properties might be empty in case of exceptional scenarios, e.g. interworking with non-oneM2M device models.

## Device Management

### Introduction

The entities that are specified in this section allow performing classical Device Management (DM) functions: rebooting a device, upgrading it, reading / setting its configuration, monitoring its logs, checking its memory or battery status, managing its firmware or its software modules, etc. They belong to the “management” domain.

In the case of a NoDN, it is the IPE in charge of exposing the device to oneM2M that creates / implements these modules. It may rely on external Device Management techniques like e.g. LwM2M (from OMA) or USP (from BBF), or any other technique, proprietary or standardized, that allows performing at least some DM functions, for instance a reboot.

The architecture of IPE-based Device Management is presented in oneM2M TS-0001 [3] clause 6.2.4.1, and the details of CRUD operations on the resources defined here are defined in oneM2M TS-0033 [21] clause 8. A developer’s guide on Device Management can be found in oneM2M TR-0035 [i.11].

### flexNode

This flexContainer specialization is the root for SDT-based Device Management modules.

The containerDefinition attribute of this specialization shall be “org.onem2m.management.device.flexNode”.

This resource is a <flexContainer> child of the <node> resource targeted by the *nodeLink* attribute of *<flexContainer>* SDT devices (see in 6.2.2 the rule 1.7).

Table 5.8.2‑1: Child resources of [*flexNode*] resource

| Child Resources of [*flexNode*] | Child Resource Type | Multiplicity | Description |
| --- | --- | --- | --- |
| *dmAreaNwkInfo\_<i>* | *[dmAreaNwkInfo]* | 0..n | See clause 5.8.10 |
| *dmAgent* | *[dmAgent]* | 0..1 | See clause 5.8.3 |
| *dmDeviceInfo* | *[dmDeviceInfo]* | 1 | See clause 5.8.4 |
| *dmDataModelIO\_<i>* | *[dmDataModelIO]* | 0..N | See clause 5.8.5 |
| *dmFirmware\_<i>* | *[dmFirmware]* | 1..N | See clause 5.8.6 |
| *dmSoftware\_<i>* | *[dmSoftware]* | 0..N | See clause 5.8.7 |
| *dmEventLog\_<i>* | *[dmEventLog]* | 0..N | See clause 5.8.8 |
| *dmPackage\_<i>* | *[dmPackage]* | 0..N | See clause 5.8.9 |
| *battery\_<i>* | *[battery]* | 0..N | See clause 5.3.10 |
| *dmCapability\_<i>* | *[dmCapability]* | 0..N | See clause 5.8.12 |
| *dmStorage\_<i>* | *[dmStorage]* | 0..N | See clause 5.8.13 |

NOTES:

* the notation ‘\_<i>’ for child resources indicates that the resource name is the name of the child ModuleClass or SubDevice flexContainer, appended with an underscore ‘\_’ and an incrementing index so that it is unique in the [flexNode] children (e.g. “dmFirmware\_0”, “dmFirmware\_1”, etc.). The index shall not have leading 0’s.
* the current list of modules for Device Management is not fixed and can evolve with new optional features.

### dmAgent

This ModuleClass is the entry point module of [*flexNode*]; it provides capabilities to control and monitor the Device Management of the device.

Table 5.8.3-1 Actions of dmAgent ModuleClass

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Return Type | Name | Arguments | Optional | Description |
| none | reboot | rebootType: hd:enumRebootType | false | Execute a reboot or a factory reset |
| M2MID | deployPackage | name: xs:string  version: xs:string  url: xs:url | true | Create a dmPackage.  Return the ID of the created package. |

The *deployPackage* action allows creating a new [dmPackage] module class (see clause 5.8.9), child of this dmAgent’s parent *flexNode*. The returned value is the ID of this created <flexContainer>. The created dmPackage is in NotInstalled state.

The DataPoints of dmAgent Module Class are as follows:

‘state’ represents the state of the agent for DM purposes (ready, sleeping, etc.).

some optional device properties which can be used for Device Management purpose. The dmAgent can be seen as a ‘dashboard’ that gathers common information such as battery level, memory or CPU usage…

Table 5.8.3-2 DataPoints of dmAgent ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Description |
| state | hd:enumDmAgentState | R | false |  | The current state of the agent (ready, error, etc.) |
| storageAvailable | xs:integer | R | true | KB | The size of available storage memory. |
| storageTotal | xs:integer | R | true | KB | The size of total storage memory. |
| ramAvailable | xs:integer | R | true | KB | The size of available RAM memory. |
| ramTotal | xs:integer | R | true | KB | Total size of the RAM memory. |
| powerStatus | hd:enumPowerState | R | true |  | The status of the electrical power. |
| cpuUsage | xs:integer | R | true | pct | Current CPU usage in percent. |
| systemTime | m2m:timestamp | RW | true |  | Reference time for the device. |

### dmDeviceInfo

This ModuleClass is used to share static information regarding the device.

Table 5.8.4-1 DataPoints of dmDeviceInfo ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Description |
| serialNumber | xs:string | R | true |  | Unique device label assigned by the manufacturer.  The value of the datapoint typically exposes the device’s serial number that is specific to a manufacturer. |
| manufacturer | xs:string | R | true |  | The name/identifier of the device manufacturer. |
| manufacturerDetailsLink | xs:anyURI | RW | true |  | URL to manufacturer’s website. |
| manufacturingDate | m2m:timestamp | R | true |  | Manufacturing date of device. |
| model | xs:string | R | true |  | The name/identifier of the device model assigned by the manufacturer. |
| subModel | xs:string | R | true |  | Device sub-model name. |
| hwVersion | xs:string | R | true |  | The hardware version / revision of the device. |
| osVersion | xs:string | R | true |  | Version of the operating system (defined by manufacturer). |
| country | m2m:countryCode | R | true |  | Country code of the device. It could be manufacturing country, deployment country or procurement country. |
| supportURL | xs:anyURI | RW | true |  | URL that points to product support information of the device. |
| presentationURL | xs:anyURI | RW | true |  | To quote UpnP: “the control point can retrieve a page from this URL, load the page into a web browser, and depending on the capabilities of the page, allow a user to control the device and/or view device status. The degree to which each of these can be accomplished depends on the specific capabilities of the presentation page and device”. |
| friendlyName | xs:string | RW | true |  | The device friendly name. |
| description | xs:string | RW | true |  | A human readable description of the device (e.g. Alice’s cell phone, kitchen’s fridge…) |

NOTE: although all datapoints are optional, depending on the underlying DM technology, some datapoints should be filled, for instance serialNumber, manufacturer and model when this information is available.

### dmDataModelIO

This ModuleClass provides capabilities to handle the device’s Data Model for cases where the underlying Device Management technology supports APIs that are not directly reflected in the *flexNode* modules.

Table 5.8.5-1 Actions of dmDataModelIO ModuleClass

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Return Type | Name | Arguments | Optional | Description |
| xs:string | readIO | address: xs:string | true | Read the current values of parameters. Argument: the list of the parameter names.  Returns a JSON serialization of the parameters (see Rules 3-2 and 3-6 in clause 6.2.4). |
| xs:string | writeIO | address: xs:string  payload: xs:string | true | Update the current values of parameters. Arguments:  ‘address’: the list of the parameter names,  ‘payload’: the list of the parameter values.  Returns the list of the modified parameter names. |

Table 5.8.5-2 DataPoints of dmDataModelIO ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Description |
| dataModelType | hd:enumDataModelType | R | false |  | The type of the data model (OMA DM, OMA LwM2M, BBF TR-181, etc.). |

The *readIO* and *writeIO* actions are defined for handling parameters of the underlying Device Management protocol using the APIs defined by those technologies. The values used in the *address* argument are dependent on the value of the *dataModelType* data point of the *dmAgent* module.

Example of *address* values could be ‘Device.WiFi.SSID.1’ if *dataModelType*=6 (BBF TR-181 USP) or ‘/9/1/1’ if *dataModelType*=4 (OMA LwM2M).

These *address* and *payload* argument can contain several values separated by a comma ‘,’. The *payload* argument’s value types are the valid JSON primitive types (string, number, boolean, null).

Examples:

readIO(address=”Device.WiFi.SSID.SSID”)

* {“Device.WiFi.SSID.SSID”: “SSIDName”}

readIO(address=”/3/0/1,/3/0/2”)

* {“/3/0/1”:“MyCompany”,“/3/0/2”:“SN376575A86”}

writeIO(address=”Device.DeviceInfo.FriendlyName,Device.WiFi.SSID.SSID”,

payload=”my device,my ssid”}

* {“Device.DeviceInfo.FriendlyName,Device.WiFi.SSID.SSID”}

writeIO(address=”/3/0/15”, payload=”[Europe/Paris](https://en.wikipedia.org/w/index.php?title=Europe/Paris&action=edit&redlink=1)”)

* {“/3/0/15”}

NOTE: some datapoints of the *dmAgent* and *dmDeviceInfo* moduleClasses correspond to fixed parameters in OMA & BBF data models. The corresponding concepts in OMA DM / LwM2M data models (resp. BBF TR-181) are specified in oneM2M TS-0005 (resp. TS-0006). For instance the datapoint memAvailable corresponds to ‘Device.DeviceInfo.MemoryStatus.Free’ in TR-181 (see TS-0006 clause 7.3) and to ‘/3/0/10’ in LwM2M (TS-0005 clause 6.3.4).

### dmFirmware

This ModuleClass provides Device Management capabilities to control and monitor the firmware of a device.

The device can contain multiple components (a graphic card for instance) that can have individual firmwares, and they need to be managed separately. The [*flexNode*] allows one [*dmFirmware*] module per component plus one ‘major’ [*dmFirmware*] for the device itself.

Individual firmwares are managed using the [*dmFirmware*] actions presented in Table 5.8.6-1.

Table 5.8.6-1 Actions of dmFirmware ModuleClass

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Return Type** | **Name** | **Argument** | **Optional** | **Description** |
| xs:string | updateFirmware | url: xs:url  version: xs:string | true | Downloads a new firmware to the device / sub-component. In case of devices that do support toggling between multiple preinstalled firmware versions it also starts the firmware flashing/installation process.  The updateFirmware action as it results returns an AE/IPE message indicating if the action was successful or not. |
| xs:string | toggle | none | true | Toggles between the firmware versions installed on a device/sub-component. In case of devices that do not support such toggling, it triggers the firmware flashing/installation process.  The toggle action as it results returns an AE/IPE message indicating if the action was successful or not. |

The abstraction model used for [dmFirmware] manages the firmware through two images: a *primary* firmware image and a *secondary* one. Despite the naming both images are equivalent and a secondary image can be actively used by a device just like the primary one.

Using an abstraction model based on two firmware images it is possible to effectively manage firmware on devices with different firmware capabilities. The state machine for firmware management using two images is shown in Figure 5.8.6-1 for devices that do support toggling between multiple preinstalled firmware versions and in Figure 5.8.6-2 for devices that can have only one firmware version installed.

**Table 5.8.6-2 DataPoints of dmFirmware ModuleClass**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Type** | **R/W** | **Optional** | **Unit** | **Description** |
| multiFirmware | xs:boolean | R | false |  | Indicates if the device/sub-component supports toggling between multiple preinstalled firmware versions |
| primaryState | hd:enumFirmwareState | R | false |  | The current state of the primary firmware image (active, downloading, etc.) |
| primaryName | xs:string | R | false |  | The name of the primary firmware image. |
| primaryVersion | xs:string | R | false |  | The version of the primary firmware image. |
| primaryUrl | xs:url | R | true |  | The URL from which the primary firmware image was downloaded |
| secondaryState | hd:enumFirmwareState | R | true |  | The current state of the secondary firmware image (active, downloading, etc.). Mandatory when updateFirmware is available |
| secondaryName | xs:string | R | true |  | The name of the secondary firmware image. |
| secondaryVersion | xs:string | R | true |  | The version of the secondary firmware image. |
| secondaryUrl | xs:url | R | true |  | The URL from which the secondary firmware image was downloaded |
| component | xs:string | R | true |  | Allows to identify the sub-component that uses this firmware.  This datapoint is mandatory if this is a sub-component firmware. |

NOTE: both primary and secondary firmware image related dataPoints are mandatory when updateFirmware is available, however depending on the device capabilities one of the two state machines – the one presented in Figure 5.8.6-1 or the one shown in Figure 5.8.6-2 should be used.

Ready

Installing

Downloading

Installing

Downloading

Ready

Active

PRIMARY

SECONDARY

TOGGLE

Active

Failure

Failure

Figure 5.8.6-1: Lifecycle of a dmFirmware for devices that support toggling between preinstalled firmware images

For devices that support toggling between multiple preinstalled firmware images the following rules apply:

- There is always one firmware image that is in “Active” state.

- Toggling between firmware images is only possible if one image is in “Ready” state and the other image is in “Active” state.

- *updateFirmware* action is always performed on the image that is in “Ready” or “Failure” state.

NOTE: it is the AE/IPE responsibility to provide the appropriate action result pointing if the action was triggered on the device or not (e.g. if the current firmware state did not allow it).

Archival

Active

Downloading

Active

Downloading

Archival

Installing

PRIMARY

SECONDARY

TOGGLE

Installing

Ready

Ready

Failure

Failure

Figure 5.8.6-2: Lifecycle of a dmFirmware for devices that can have only one firmware version installed

In case of a device that can have only one firmware version installed the additional firmware image is used differently. First of all it’s treated as a temporary storage for the *updateFirmware* action that triggers the download process. When the download process is finished the status of the firmware image is changed to ”Ready”. It’s important to note that in this case the installation/flashing process is started after the *toggle* action is issued, making it a “long toggle”. At the very same moment the toggle is issued, the previously active firmware image is moved to “Archival” state, making it a historical record that cannot be restored. The reason why it cannot be restored is trivial – it was just overwritten by the flashing process.

For devices that can have only one firmware version installed the following rules apply:

* There is always no more than one firmware image that is in “Active” or “Installing” state.
* Toggling between firmware images is only possible if one image is in “Ready” state and the other image is in “Active” state.
* The *toggle* action moves the image that was previously in “Active” state to the “Archival” state.
* The *toggle* action moves the image that was previously in “Ready” state to the “Installing” state.
* *updateFirmware* action is always performed on the image that is in “Archival” or “Failure” state.
* If one image is in “Archival” state and the other is in “Failure” state the *updateFirmware* action is always performed on the image that is in “Archival” state.

NOTE: it is the AE/IPE responsibility to provide the appropriate action result pointing if the action was triggered on the device or not (e.g. if the current firmware state did not allow it).

### dmSoftware

This ModuleClass provides DM capabilities to control and monitor software modules of the device.

An instance of this module class represents a software module hosted by the device.

A [dmSoftware] module is created on a Hosting CSE by the IPE in charge of the device, either at the initialization if it represents a software module that is pre-installed on the device, or after installation of one or more [dmPackage] modules (see clause 5.8.9) that have been dynamically created (for instance a software image with associated configuration files and libraries).

The association between one or more dmPackage modules and a dmSoftware module are under the responsibility of the IPE: dmSoftware modules are created, deleted or updated only by the IPE (for instance updating a dmPackage can trigger the modification of the *version* datapoint of an associated dmSoftware).

From external applications, [dmSoftware] modules can only be discovered from the parent [flexNode], not created, and afterwards they can only be activated / deactivated. They can be seen as ‘high level’ information (“there is such software that is running on the device”), whereas dmPackages are ‘low level’ information (“there is such executable file that is deployed on the device”).

Table 5.8.7-1 Actions of dmSoftware ModuleClass

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Return Type | Name | Argument | Optional | Description |
| none | activate | none | true | Activate the software module. |
| none | deactivate | none | true | Deactivate the software module. |

Table 5.8.7-2 DataPoints of dmSoftware ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Description |
| state | hd:enumSoftwareState | R | false |  | The current state of the software module (see clause 5.6.47). |
| name | xs:string | R | true |  | The name of the software module. |
| version | xs:string | R | true |  | The version of the software module. |

### dmEventLog

This ModuleClass provides DM capabilities to control and monitor event logs of the device.

Table 5.8.8-1 Actions of dmEventLog ModuleClass

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Return Type | Name | Arguments | Optional | Description |
| none | retrieveLog | start: xs:datetime  end: xs:datetime | true | Upload from the device the logging data between ‘start’ and ‘end’.  ‘start’ must be a date before ‘end’, and is optional. The default is beginning of time.  ‘end’ must be a date after ‘start’ and is optional. The default is the timestamp of the last available log entry. |

This action, if provided, requests the IPE to read logging data on the device. This log is then stored in the ‘data’ datapoint. It is only valid when the ‘enabled’ datapoint is *true*. The *start* and *end* arguments are only indications of the timeframe for the log retrieval. If a target device can deliver only partial logs for a given timeframe, for example when the *start* argument is too far in the past and logs are not available for that time anymore, then the device shall deliver logs from the earliest available point in time on.

Table 5.8.8-2 DataPoints of dmEventLog ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Description |
| type | m2m:logTypeId | R | false |  | The type of the log (e.g. security log, system log…). |
| data | xs:string | R | false |  | Raw data of *last* event. No format specified. |
| status | m2m:logStatus | R | false |  | The current status of the logging process (Started, Stopped, Error, etc.) |
| enabled | xs:boolean | RW | false |  | Start / stop logging. |

For devices using the dmEventLog ModuleClass, the following rules apply:

* The actual logging process on the device (if any), and the retrieval of device logging data by the IPE, are out of scope of this document.
* Instances of this module should only be created by the IPE (one per log type supported by the device for instance).
* The IPE can create a [dmEventLog] instance with *status* datapoint ‘NotPresent’ for a given log type, to indicate that this log type is not supported by the device. Otherwise *status* should have value ‘Started’ (resp. ‘Stopped’) if the *enabled* datapoint is set to *true* (resp. *false*). The *status* datapoint can be given ‘Error’ value if the log processing dysfunctions.
* The IPE should use the <*flexContainerInstance*> history mechanism (see TS-0001 § 9.6.59) by setting on [dmEventLog] at least one attribute *maxNrOfInstances*, *maxByteSize* or *maxInstanceAge*. Then for each log event read by the IPE from the device, and if the *enabled* datapoint has value *true*, a <*flexContainerInstance*> resource shall be created, child of this module <*flexContainer*>. The [dmEventLog] module itself just contains the *last* logged event from the device for this log type.
* The [dmEventLog] <*flexContainer*>, and therefore its <*flexContainerInstance*> children resources, should have a *dataGenerationTime* custom attribute that indicates the time the event was logged *on the device* (see Rule 2-5 in section 6.2.3).

When the *enabled* datapoint is set to *false*, the IPE shall set the *status* datapoint to ‘Stopped’ and shall not modify the *data* datapoint of the module, and therefore shall not create any <*flexContainerInstance*> child resource.

### dmPackage

This ModuleClass provides DM capabilities to deploy, control and monitor packages of the device.

* These packages can be simple resource files such as software libraries, configuration files, etc. In this case the *softwares* datapoint will be empty.
* They also can correspond to software images, in which case their installation will trigger the creation by the IPE of one or more [dmSoftware] SDT modules classes that can be activated / deactivated (see clause 5.8.7). In this case the *softwares* datapoint will contain the list of IDs of this(these) dmSoftware module(s).
* Instances of the dmPackage module class can be dynamically created by the *deployPackage* action of the dmAgent module class (see clause 5.8.2).

Table 5.8.9-1 Actions of dmPackage ModuleClass

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Return Type | Name | Argument | Optional | Description |
| none | install | none | false | Download if needed and install the package. |
| none | uninstall | none | false | Uninstall the package. |
| none | update | version: xs:string  url: xs:string | false | Update the package. |

Notes:

* The package can be pre-downloaded when the [dmPackage] resource is created.
* In the *update* action, the arguments *version* and *url* can be empty strings (case for instance of updating a package on a Linux-type system).
* When the [dmPackage] resource is deleted, the package shall be removed from the device.

Table 5.8.9-2 DataPoints of dmPackage ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Description |
| type | hd:enumPackageType | R | false |  | The type of the package (software, library, config file, web content, etc. See clause 5.6.49) |
| state | hd:enumPackageState | R | false |  | The current state of the package (see clause 5.6.48) |
| name | xs:string | R | true |  | The name of the package. |
| version | xs:string | R | true |  | The version of the package. |
| url | xs:url | R | true |  | The URL from which the package can be downloaded |
| softwares | m2m:listOfM2MID | R | true |  | The list of dmSoftware modules, if any, that are associated with this dmPackage |

Notes:

* the dmPackage *name* and *version* datapoints are optional because they can be deduced from the downloaded resource. The *url* datapoint is optional because the package can be pre-installed or downloaded from a default repository (for instance a package on a Linux-type system).
* The possible dependencies between dmPackage modules (for instance the dmPackage of an executable software image depends on the deployment of other dmPackage that correspond to libraries needed by this software) is out of scope of this document.

The control of the association between a dmPackage and an associated dmSoftware, for instance updating a dmPackage when the dmSoftware is active, is out of scope of this document.

### dmAreaNwkInfo

A dmAreaNwkInfo is a SDT SubDevice entity, mapped as a <flexContainer> resource that expresses the information about the devices in a M2M Area Network managed by the parent flexNode.

Table 5.8.10-1: Properties of dmAreaNwkInfo model

|  |  |  |  |
| --- | --- | --- | --- |
| Property Name | Property Type | Multiplicity | Description |
| propAreaNwkType | xs :string | 1 | Indicates the type of M2M Area Network |

Table 5.8.10-2: Modules of dmAreaNwkInfo model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| dmAreaNwkDeviceInfo | dmAreaNwkDeviceInfo | 0..N | See clause 5.8.11 |

### dmAreaNwkDeviceInfo

This ModuleClass is used to share information regarding the devices in the M2M Area Network.

Table 5.8.11-1 DataPoints of dmAreaNwkDeviceInfo ModuleClass

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | R/W | Optional | Unit | Description |
| devId | xs:string | R | false |  | Indicates the id of the device. It could be the id of the hardware or nodeId. |
| devType | xs:string | R | false |  | Indicates the type of the device. The attribute also indicates the functions or services that are provided by the device. Examples include temperature sensor, actuator, Zigbee coordinator or Zigbee router |
| sleepInterval | xs:integer | R | true | seconds | The interval between two sleeps. |
| sleepDuration | xs:integer | R | true | seconds | The time duration of each sleep. |
| status | xs:string | R | true |  | The status of the device (sleeping or waked up). |

### dmCapability

This ModuleClass is used to model the service capabilities of a managed device.

Table 5.8.12-1: Actions of dmCapability ModuleClass

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Return Type | Name | Argument | Optional | Documentation |
| m2m:status | enable | none | true | The action that allows enabling the device capability.  Returns the status of the action. |
| m2m:status | disable | none | true | The action that allows disabling the device capability.  Returns the status of the action. |

Table 5.8.12-2 DataPoints of dmCapability ModuleClass

| Name | Type | R/W | Optional | Unit | Description |
| --- | --- | --- | --- | --- | --- |
| name | xs:string | R | false |  | The name of the device capability. |
| attached | xs:boolean | R | false |  | Indicates whether the capability is currently attached to the device or not. |
| currentState | xs: boolean | R | false |  | Indicates the current state of the capability (e.g. enabled or disabled). |

### dmStorage

This ModuleClass is used to model the storage on a managed device.

Table 5.8.13-1: Actions of dmStorage ModuleClass

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Return Type | Name | Argument | Optional | Documentation |
| none | format | none | true | The action that allows to format the mounted storage. |
| none | unmount | none | true | The action that allows to safety eject storage device. |

Table 5.8.13-2 DataPoints of dmStorage ModuleClass

| Name | Type | R/W | Optional | Unit | Description |
| --- | --- | --- | --- | --- | --- |
| UUID | xs:string | R | true |  | The uuid of the storage device. |
| type | xs:integer | R | true |  | Indicates the type of storage. 0 indicates internal and 1 indiciates external. |
| name | xs:string | RW | true |  | Indicates name of the mounted storage. |
| writeSpeed | xs:integer | R | true |  | Indicates the write speed of storage device. |
| readSpeed | xs:integer | R | true |  | Indicates the read speed of storage device. |
| availStorage | xs:integer | R | false | MB | Indicates the current available amount of memory. |
| totalStorage | xs:integer | R | false | MB | Indicates the total amount of memory available. |
| presence | xs:integer | R | true |  | Indicates current presence status of memory card. 0 indicates card is ejected, 1 indicates card is inserted. |
| status | xs:integer | R | true |  | Indicates current operation status of storage. 1 –indicates storage is ready, 0 indicates storage is busy. |
| mounts | xs:integer | R | true |  | Indicates number of successful mounts of the storage. |
| forcedUnmounts | xs:integer | R | true |  | Indicates number of forced unmounts of the storage. |
| fileSystem | xs:string | RW | true |  | Indicates the filesystem type used on the mounted storage. |
| mountingPoint | xs:string | RW | true |  | Indicates mounting point of the mounted storage. |
| mountOptions | xs:string | R | true |  | Indicates additional file system specific and file system independent mount options that indicate specific behaviours of the mount point as well as the capabilities of the underlying file system. |
| writable | xs:boolean | R | false |  | Indicates whether the storage volume is mounted as read/write (“TRUE”) or read-only (“FALSE”). |

# The Principle of Resource Mapping for Home Appliance Information Model

## Introduction

Home appliance information models which are defined in clause 5need to be represented as resources in the oneM2M system. This clause defines the principle of resource mapping based on <flexContainer>. The individual information mapping is provided in annexes A, B, C and D.

## The Resource Mapping Rules

### Introduction

The present clause specifies the rule to map the "Harmonized Information Model" to oneM2M resources.

### Resource mapping for Device model

When the AE exposes a controlling interface for a home domain device which is specified as an information model in clause 5.5, a specialization of the <flexContainer> resource shall be created as the mapping of the model following conversion rules:

* Rule 1-1: Each Device model defined in clause 5.5 shall be mapped to a specialization of <flexContainer>. The *containerDefinition* attribute shall be set according to 6.4.2.
* Rule 1-2: Each entry in the 'Module' table shall be mapped to a child resource(s) which is mapped as a specialised <flexContainer> following the rule in clause 6.2.3.
* Rule 1-3: The specialized <flexContainer> resource of the Device model may contain an optional attribute *nodeLink* (as defined in TS-0001[3] and in TS-0004[4]). The value of *nodeLink* shall be set to the resource identifier of a <node> resource described in Rule 1-5 below. See also Rule 1-8.
* Rule 1-4: XSD file for each Device model shall be named according to 6.5.2.
* Rule 1-5: A <node> resource shall be created on the same hosting CSE as the <flexContainer> representing this Device model. If the <node> resource does not contain a [*flexNode*] child resource (see Rule 1.7), then it contains all the management information as specialized <mgmtObj> resources (e.g. [firmware]) about the Device model instance for device management purposes.
* Rule 1-6: Void.
* Rule 1-7:The <node> resource targeted by the *nodeLink* attribute may have a [*flexNode*] child resource. This [*flexNode*] resource contains all the Device Management information as specialized <flexContainer> resources defined in 5.8 (e.g. [*dmFirmware*]) about the device model instance for Device Management purposes.
* Rule 1-8: Void.
* Rule 1-9: Each entry in the 'SubDevice' table shall be mapped to a child resource(s) which is mapped as a specialised <flexContainer> following the rule in clause 6.2.7.
* Rule 1-10: Each <flexContainer> associated to a Device model may have as child resource any <flexContainer> associated to a ModuleClass model of the Metadata domain defined in clause 5.3.9.

In other words, all devices implicitly have the following lines in their modules table:

Table 6.2.2-1: Modules of deviceXXX model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| <any module in mdd domain> | <any module in mdd domain> | 0..N | See clauses 5.3.9. |

### Resource mapping for ModuleClass

The ModuleClass models shall be mapped to the specializations of a <flexContainer> resource. The following rules shall be applied:

When the Device or SubDevice models in clauses 5.4, 5.5, 5.8.2 or 5.8.10 are mapped to the <flexContainer> resource, and if the device or sub-device supports the functionality associated with a ModuleClass in the model, a <flexContainer> resource which is mapped from ModuleClass definitions shall be created as a child resource:

* Rule 2-1: The containerDefinition attribute shall be set according to 6.4.3.
* Rule 2-2: Each entry of 'Action', 'Property', and 'DataPoint' in ModuleClass definitions shall be mapped following the resource mapping rules described in clauses 6.2.4 - 6.2.7.
* Rule 2-3: XSD file for each ModuleClass shall be named according to 6.5.3.
* Rule 2-4: The *resourceName* attribute for each module class that appears as a child of a Device or SubDevice model shall be CREATED with the value set to “Module Instance Name”. If the module class is contained in a list (multiplicity 0..N or 1..N), its *resourceName* attribute shall be set to “Module Instance Name” appended with an underscore ‘\_’ and an incrementing index so that it is unique in the parent’s children (e.g. “firmware\_0”, “firmware\_1”, etc.). The index shall not have leading 0’s.
* Rule 2-5: The specialized <flexContainer> resource of the Module model may contain an optional [customAttribute] named *dataGenerationTime*. The value of *dataGenerationTime* contains the time when the data was generated by the device. The data type of this custom attribute is m2m:timestamp.

### Resource mapping for Action

Actions defined as part of a ModuleClass model shall be mapped to the specializations of a <flexContainer> resource. The following rules shall be applied:

* Rule 3-1: The *containerDefinition* attribute shall be set according to 6.4.4.
* Rule 3-2: When the Action supports any 'Arguments', they are mapped to [customizedAttribute] with their variable names (short names are given in clause 6.3.4). When the Action supports a 'Return Type', it is mapped to a [customizedAttribute] named ‘result’ (short name ‘resut’). The keyword ‘result’ is reserved and cannot be used as an Argument name.
* Rule 3-3: XSD file for each Action shall be named according to 6.5.4.
* Rule 3-4: The Action shall be triggered:
  + by updating at least one of the Arguments custom attributes with any value, if the action has at least one argument, or
  + by updating the <flexContainer> resource with *empty content* if it has no argument
* Rule 3-5: The *resourceName* attribute for each Action model that appears as a child of a ModuleClass model shall be CREATED with the value set to “Action name”.
* Rule 3-6: If an action returns a value that is of a complex data type, i.e. not one of the standard scalar types, then this value shall be encoded as a JSON structure and returned serialized in an xs:string.

### Resource mapping for Property

When the Device model (in clause 5.5) or the ModuleClass model (in clause 5.3) is mapped to the <flexContainer> resource, and if the device supports a Property, the following rules shall be applied:

* Rule 4-1: Each entry of ‘Property’ table in ModuleClass model, shall be mapped to the [customAttribute] of <flexContainer> resource which is mapped from associated ModuleClass model, with its Property name with prefix 'prop'.
* Rule 4-2: If the <node> resource targeted by the *nodeLink* attribute of a Device model does not have a [*flexNode*] child resource, then each ‘Property’ of the Device model is mapped to a specialized [objectAttribute] of a [deviceInfo] <mgmtObj> resource child of this <node>, otherwise it is mapped to a [customAttribute] of a [dmDeviceInfo] <flexContainer> resource child of this [*flexNode*].
* Rule 4-3: Each entry of ‘Property’ table in SubDevice model, shall be mapped to the [customAttribute] of <flexContainer> resource which is mapped from associated SubDevice model, with its Property name with prefix 'prop'.

### Resource mapping for DataPoint

When the ModuleClass model (in clause 5.3) is mapped to the <flexContainer> resouce, and if the ModuleClass supports a DataPoint, the following rules shall be applied:

* Rule 5-1: Each entry of DataPoint table in ModuleClass model, shall be mapped to [customAttribute] of <flexContainer> resource which is mapped from associated ModuleClass model, with its DataPoint name.

### Resource mapping for SubDevice model

The SubDevice models (in clause 5.4 or 5.8.10) shall be mapped to the specializations of a <flexContainer> resource. The following rules shall be applied:

When the SubDevice model in clause 5.4 or 5.8.10 is mapped to the <flexContainer> resource, and if the device supports the functionality associated with a SubDevice in the model, a <flexContainer> resource which is mapped from SubDevices definitions shall be created as a child resource.

* Rule 7-1: The containerDefinition attribute shall be set according to 6.4.5.
* Rule 7-1a: Each entry in the 'Module' table shall be mapped to a child resource(s) which is mapped as a specialised <flexContainer> following the rule in clause 6.2.3.
* Rule 7-2: The XSD file for each SubDevice model shall be named according to clause 6.5.5.
* Rule 7-3: void
* Rule 7-4: The *resourceName* attribute for each SubDevice that appears as a child of a Device or FlexNode model shall be created with the value set to “SubDevice Instance Name”. If the SubDevice is contained in a list (multiplicity 0..N or 1..N), its *resourceName* attribute shall be set to “SubDevice Instance Name” appended with an underscore ‘\_’ and an incrementing index so that it is unique in the parent’s children (e.g. “cuff\_0”, “cuff\_1”, etc.). The index shall not have leading 0’s.
* Rule 7-5: Each <flexContainer> associated to a SubDevice model may have as child resource any <flexContainer> associated to a ModuleClass model of the Metadata domain defined in clause 5.3.9.

In other words, all subdevices implicitly have the following lines in their modules table:

Table 6.2.7-1: Modules of subDeviceXXX model

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Multiplicity | Description |
| <any module in mdd domain> | <any module in mdd domain> | 0..N | See clauses 5.3.9. |

## Short names

### Introduction

XML and JSON representations require the explicit encoding of the names of resource attributes, (in the case of XML) and resource types. Whenever a protocol binding transfers such a name over a oneM2M reference point, it shall use a shortened form of that name. Short names enable payload reduction on involved telecommunication interfaces.

The mapping between the full names and their shortened form is given in the clauses that follow.

### Resource types

In protocol bindings resource type names for device models shall be translated into short names of Table 6.3.2‑1.

Table 6.3.2‑1: Specialization type short names (Device models)

| Resource Type Name | Short Name |
| --- | --- |
| device3DPrinter | ***dTDPr*** |
| deviceAirConditioner | ***deACr*** |
| deviceAirHeater | ***deAHr*** |
| deviceAirPurifier | ***deAPr*** |
| deviceAirQualityMonitor | ***dAQMr*** |
| deviceAudioReceiver | ***deARr*** |
| deviceBloodPressureMonitor | ***dBPMr*** |
| deviceCamera | ***devCa*** |
| deviceClothesDryer | ***deCDr*** |
| deviceClothesWasher | ***deCWr*** |
| deviceClothesWasherDryer | ***dCWDr*** |
| deviceCoffeeMachine | ***deCMe*** |
| deviceCookerHood | ***deCHd*** |
| deviceCooktop | ***devCp*** |
| deviceDehumidifier | ***devDr*** |
| deviceDishWasher | ***deDWr*** |
| deviceDoor | ***devD0*** |
| deviceDoorLock | ***deDLk*** |
| deviceElectricMotorcycle | ***deEMe*** |
| deviceElectricVehicleCharger | ***dEVCr*** |
| deviceFan | ***devFn*** |
| deviceFoodProbe | ***deFPe*** |
| deviceFreezer | ***devFr*** |
| deviceGlucosemeter | ***devGr*** |
| deviceHeartRateMonitor | ***dHRMr*** |
| deviceHomeCCTV | ***dHCCT*** |
| deviceHumidifier | ***devHr*** |
| deviceJuicer | ***devJr*** |
| deviceKettle | ***devKe*** |
| deviceLight | ***devLt*** |
| deviceMicrogeneration | ***devMn*** |
| deviceMultiFunctionPrinter | ***dMFPr*** |
| deviceOutdoorLamp | ***deOLp*** |
| deviceOven | ***devOn*** |
| devicePrinter | ***devPr*** |
| devicePulseOximeter | ***dePOr*** |
| deviceRefrigerator | ***devRr*** |
| deviceRobotCleaner | ***deRCr*** |
| deviceScanner | ***devSr*** |
| deviceSecurityPanel | ***deSPl*** |
| deviceSetTopBox | ***dSTBx*** |
| deviceShoesWasher | ***deSWr*** |
| deviceSmartElectricMeter | ***dSEMr*** |
| deviceSmartPlug | ***deSPg*** |
| deviceSteamCloset | ***deSCt*** |
| deviceStorageBattery | ***deSBy*** |
| deviceSwitch | ***devSh*** |
| deviceTelevision | ***devTn*** |
| deviceThermometer | ***devTr*** |
| deviceThermostat | ***devTt*** |
| deviceWaterHeater | ***deWHr*** |
| deviceWaterValve | ***deWVe*** |
| deviceWeightScaleAndBodyCompositionAnalyser | ***dWSAB*** |
| deviceWindowShade | ***deWSe*** |
| deviceBottleWarmer | ***deBWr*** |
| deviceGarbageDisposal | ***deGDp*** |
| deviceWaterPurifier | ***deWPr*** |
| flexNode | ***fleNe*** |

In protocol bindings resource type names for SubDevice model shall be translated into short names of Table 6.3.2‑2.

Table 6.3.2‑2: Specialization type short names (SubDevice models)

| Resource Type Name | Short Name |
| --- | --- |
| cuff | ***cuff*** |
| powerOutlet | ***powOt*** |
| subDeviceCuff | ***suDCf*** |
| subDevicePowerOutlet | ***sDPOt*** |
| dmAreaNwkInfo | ***dANIo*** |

NOTE: see section 6.2.7, rule 7-4

In protocol bindings resource type names for module classes shall be translated into short names of Table 6.3.2‑3.

Table 6.3.2‑3: Specialization type short names (ModuleClasses and Module Instances)

| Resource Type Name | Short Name |
| --- | --- |
| 3DPrinter | ***thDPr*** |
| acousticSensor | ***acoSr*** |
| airCleanOperationMode | ***aCOM0*** |
| airConJobMode | ***aCJMe*** |
| airConOperationMode | ***aCOMe*** |
| airFlow | ***airFw*** |
| airPurifierJobMode | ***aPJMe*** |
| airPurifierOperationMode | ***aPOMe*** |
| airQualitySensor | ***aiQSr*** |
| alarmSpeaker | ***alaSr*** |
| audioVolume | ***audVe*** |
| autoDocumentFeeder | ***auDFr*** |
| battery | ***bat*** |
| binaryObject | ***binOt*** |
| binarySwitch | ***binSh*** |
| bioElectricalImpedanceAnalysis | ***bEIAs*** |
| bodyCompositionAnalyser | ***boCAr*** |
| boiler | ***boilr*** |
| boilingSwitch | ***boiSh*** |
| brewing | ***brewg*** |
| brewingSwitch | ***breSh*** |
| brightness | ***brigs*** |
| channel | ***chanl*** |
| clock | ***clock*** |
| clothesDryerJobMode | ***cDJMe*** |
| clothesDryerOperationMode | ***cDOMe*** |
| clothesWasherDryerJobMode | ***cWDJM*** |
| clothesWasherDryerOperationMode | ***cWDOM*** |
| clothesWasherJobMode | ***cWJMe*** |
| clothesWasherJobModeOption | ***cWJMO*** |
| clothesWasherOperationMode | ***cWOMe*** |
| colour | ***color*** |
| colourSaturation | ***colSn*** |
| controlPanelLock | ***coPLk*** |
| cookerHoodJobMode | ***cHJMe*** |
| credentials | ***creds*** |
| customTemperature | ***cusTe*** |
| dataGenerationTime | ***dgt*** |
| dehumidifierJobMode | ***deJMe*** |
| dehumidifierOperationMode | ***deOMe*** |
| dishWasherJobMode | ***dWJMe*** |
| dmAgent | ***dmAgt*** |
| dmAreaNwkDeviceInfo | ***dANDo*** |
| dmCapability | ***dmCay*** |
| dmDataModelIO | ***dDMIO*** |
| dmDeviceInfo | ***dmDIo*** |
| dmEventLog | ***dmELg*** |
| dmFirmware | ***dmFie*** |
| dmPackage | ***dmPae*** |
| dmSoftware | ***dmSoe*** |
| dmStorage | ***dmSte*** |
| doorLock | ***dooLk*** |
| doorStatus | ***dooSs*** |
| electricVehicleConnector | ***elVCr*** |
| energyConsumption | ***eneCn*** |
| energyGeneration | ***eneGn*** |
| faultDetection | ***fauDn*** |
| features | ***feats*** |
| filterInfo | ***filIo*** |
| foaming | ***foamg*** |
| fridgeTemperature | ***friTe*** |
| frozenTemperature | ***froTe*** |
| geoLocation | ***geoLn*** |
| glucometer | ***glucr*** |
| grinder | ***grinr*** |
| heatingZone | ***heaZe*** |
| height | ***heigt*** |
| hotWaterSupply | ***hoWSy*** |
| impactSensor | ***impSr*** |
| keepWarm | ***keeWm*** |
| keypad | ***keypd*** |
| liquidLevel | ***liqLl*** |
| liquidRemaining | ***liqRg*** |
| location | ***locan*** |
| localization | ***loca0*** |
| lock | ***lock*** |
| mediaInput | ***medIt*** |
| mediaOutput | ***medOt*** |
| mediaSelect | ***medSt*** |
| milkFoaming | ***milFg*** |
| milkQuantity | ***milQy*** |
| milkStatus | ***milSs*** |
| motionSensor | ***motSr*** |
| numberValue | ***numVe*** |
| openLevel | ***opeLl*** |
| operationMode | ***opeMe*** |
| origin | ***orign*** |
| overcurrentSensor | ***oveSr*** |
| oximeter | ***oximr*** |
| ozoneMeter | ***ozoMr*** |
| phoneCall | ***phoCl*** |
| playerControl | ***plaCl*** |
| powerSave | ***powS0*** |
| printerRunState | ***prRSe*** |
| printQueue | ***priQe*** |
| pulsemeter | ***pulsr*** |
| pushButton | ***pusBn*** |
| recorder | ***recor*** |
| refrigeration | ***refrn*** |
| relativeHumidity | ***relHy*** |
| remoteControlEnable | ***reCEe*** |
| robotCleanerJobMode | ***rCJMe*** |
| robotCleanerOperationMode | ***rCOMe*** |
| runState | ***runSe*** |
| scannerRunState | ***scRSe*** |
| securityMode | ***secMe*** |
| sessionDescription | ***sesDn*** |
| signalStrength | ***sigSh*** |
| sleepTimer | ***sleTr*** |
| smokeSensor | ***smoSr*** |
| sphygmomanometer | ***sphyr*** |
| spinLevel | ***spiLl*** |
| steamClosetJobMode | ***sCJMe*** |
| steamClosetOperationMode | ***sCOMe*** |
| televisionChannel | ***telCl*** |
| temperature | ***tempe*** |
| temperatureAlarm | ***temAm*** |
| textMessage | ***texMe*** |
| timer | ***timer*** |
| turbo | ***turbo*** |
| uvSensor | ***uveSr*** |
| waterFilterInfo | ***waFIo*** |
| waterFlow | ***watFw*** |
| waterSensor | ***watSr*** |
| waterStatus | ***watSs*** |
| weight | ***weigt*** |

In protocol bindings resource type names for actions shall be translated into short names of Table 6.3.2‑4.

Table 6.3.2‑4: Specialization type short names (Actions)

| Resource Type Name | Short Name |
| --- | --- |
| **activate** | actie |
| activateClockTimer | ***acCTr*** |
| answer | ***answr*** |
| call | ***call*** |
| close | ***close*** |
| deactivate | ***deace*** |
| deactivateClockTimer | ***deCTr*** |
| decrementNumberValue | ***deNVe*** |
| deployPackage | ***depPe*** |
| disable | ***disae*** |
| downChannel | ***dowCl*** |
| downVolume | ***dowVe*** |
| enable | ***enabe*** |
| format | ***formt*** |
| hangup | ***hangp*** |
| incrementNumberValue | ***inNVe*** |
| install | ***instl*** |
| nextTrack | ***nexTk*** |
| open | ***open*** |
| previousTrack | ***preTk*** |
| reboot | ***rebot*** |
| readIO | ***reaIO*** |
| resetNumberValue | ***reNVe*** |
| resetTextMessage | ***reTMe*** |
| start3Dprint | ***staDt*** |
| stop3Dprint | ***stoDt*** |
| toggle | ***togge*** |
| uninstall | ***uninl*** |
| unmount | ***unmot*** |
| upChannel | ***uphCl*** |
| updateFirmware | ***updFe*** |
| upVolume | ***upoVe*** |
| writeIO | ***wriIO*** |

### Resource attributes for properties and data points

In protocol bindings resource attributes names for properties of module classes shall be translated into short names of Table 6.3.3‑1.

Table 6.3.3‑1: Resource attribute short names (ModuleClass properties)

|  |  |  |
| --- | --- | --- |
| Attribute Name | Occurs in | Short Name |
| None |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

In protocol bindings resource attributes names for data points of module classes shall be translated into short names of Table 6.3.3‑2.

Table 6.3.3‑2: Resource attribute short names (ModuleClass data points)

|  |  |  |
| --- | --- | --- |
| Attribute Name | Occurs in | Short Name |
| absoluteEnergyConsumption | energyConsumption | ***abECn*** |
| absoluteStartTime | timer | ***abSTe*** |
| absoluteStopTime | timer | ***abST0*** |
| acousticStatus | acousticSensor | ***acoSs*** |
| adfStates | autoDocumentFeeder | ***adfSs*** |
| alarm | motionSensor, smokeSensor, temperatureAlarm, waterSensor | ***alarm*** |
| alarmStatus | alarmSpeaker | ***alaSs*** |
| altitude | geoLocation | ***altie*** |
| automode | airFlow | ***autoe*** |
| availableChannels | televisionChannel | ***avaCs*** |
| basalMetabolism | bodyCompositionAnalyser | ***basMm*** |
| bath | hotWaterSupply | ***bath*** |
| batteryThreshold | battery | ***batTd*** |
| blue | colour | ***blue*** |
| bmi | bodyCompositionAnalyser | ***bmi*** |
| bodyLength | bodyCompositionAnalyser | ***bodLh*** |
| bone | bioElectricalImpedanceAnalysis | ***bone*** |
| brightness | brightness | ***brigs*** |
| callerID | phoneCall | ***calID*** |
| callState | phoneCall | ***calSe*** |
| capacity | battery | ***capay*** |
| ch2o | airQualitySensor | ***ch2o*** |
| channelId | televisionChannel | ***chaId*** |
| channelName | televisionChannel | ***chaNe*** |
| charging | battery | ***charg*** |
| chargingCapacity | electricVehicleConnector | ***chaCy*** |
| co | airQualitySensor | ***co*** |
| co2 | airQualitySensor | ***co2*** |
| coarseness | grinder | ***coars*** |
| code | faultDetection, filterInfo | ***code*** |
| coldWash | clothesWasherJobModeOption | ***colWh*** |
| colourSaturation | colourSaturation | ***colSn*** |
| component | dmFirmware | ***compt*** |
| concentration | glucometer | ***concn*** |
| contextCarbohydratesAmount | glucometer | ***coCAt*** |
| contextCarbohydratesSource | glucometer | ***coCSe*** |
| contextExercise | glucometer | ***conEe*** |
| contextHealth | glucometer | ***conHh*** |
| contextLocation | glucometer | ***conLn*** |
| contextMeal | glucometer | ***conMl*** |
| contextMedication | glucometer | ***conMn*** |
| contextTester | glucometer | ***conTr*** |
| country | dmDeviceInfo | ***couny*** |
| cpuUsage | dmAgent | ***cpuUe*** |
| cupsNumber | brewing | ***cupNr*** |
| current | energyConsumption | ***currt*** |
| currentAdfState | autoDocumentFeeder | ***cuASe*** |
| currentDate | clock | ***curDe*** |
| currentJobMode | airConJobMode, airPurifierJobMode, clothesDryerJobMode, clothesWasherDryerJobMode, clothesWasherJobMode, cookerHoodJobMode, dehumidifierJobMode, dishWasherJobMode, robotCleanerJobMode, steamClosetJobMode | ***cuJMe*** |
| currentJobModeName | airConJobMode, airPurifierJobMode, clothesDryerJobMode, clothesWasherDryerJobMode, clothesWasherJobMode, cookerHoodJobMode, dehumidifierJobMode, dishWasherJobMode, robotCleanerJobMode, steamClosetJobMode | ***cJMNe*** |
| currentJobState | runState | ***cuJSe*** |
| currentMachineState | runState | ***cuMSe*** |
| currentPlayerMode | playerControl | ***cuPMe*** |
| currentPlayerModeName | playerControl | ***cPMNe*** |
| currentSecurityMode | securityMode | ***cuSMe*** |
| currentTemperature | temperature | ***curT0*** |
| currentTime | clock | ***curTe*** |
| currentTimeZone | clock | ***cuTZe*** |
| currentValue | smokeSensor | ***crv*** |
| data | dmEventLog | ***data*** |
| dataModelType | dmDataModelIO | ***daMTe*** |
| dataSourceID | origin | ***daSID*** |
| dataType | origin | ***datTe*** |
| defaultValue | numberValue, textMessage | ***defVe*** |
| defrost | refrigeration | ***defrt*** |
| description | faultDetection, dmDeviceInfo, localization | ***dc*** |
| desiredHumidity | relativeHumidity | ***desHy*** |
| detectedTime | overcurrentSensor, smokeSensor | ***detTe*** |
| diastolicPressure | sphygmomanometer | ***diaPe*** |
| discharging | battery | ***discg*** |
| dischargingCapacity | electricVehicleConnector | ***disCy*** |
| doorState | doorStatus | ***dooSe*** |
| duration | overcurrentSensor, recorder | ***dur*** |
| electricEnergy | battery | ***eleEy*** |
| enabled | dmEventLog | ***enabd*** |
| energy | pulsemeter | ***enery*** |
| estimatedTimeToEnd | timer | ***eTTEd*** |
| extraRinse | clothesWasherJobModeOption | ***extRe*** |
| fat | bioElectricalImpedanceAnalysis | ***fat*** |
| fatFreeMass | bodyCompositionAnalyser | ***faFMs*** |
| filterLifetime | filterInfo | ***filLe*** |
| foamingStrength | foaming | ***foaSh*** |
| frequency | energyConsumption | ***freqy*** |
| friendlyName | dmDeviceInfo | ***friNe*** |
| friendlyLocation | location, localization | ***friLn*** |
| fwVersion | dmDeviceInfo | ***fweVn*** |
| generationSource | energyGeneration | ***genSe*** |
| geoJSON | location | geoJN |
| grainsRemaining | grinder | ***graRg*** |
| green | colour | ***green*** |
| hash | binaryObject | ***hash*** |
| hba1c | glucometer | ***hba1c*** |
| heading | geoLocation | ***headg*** |
| headingAccuracy | geoLocation | ***heaAy*** |
| heatingLevel | heatingZone | ***heaLl*** |
| height | height | ***heigt*** |
| horizontalAccuracy | geoLocation | ***horAy*** |
| horizontalDirection | airFlow | ***horDn*** |
| hwVersion | dmDeviceInfo | ***hweVn*** |
| impactDirectionHorizontal | impactSensor | ***imDHl*** |
| impactDirectionVertical | impactSensor | ***imDVl*** |
| impactLevel | impactSensor | ***impLl*** |
| impactStatus | impactSensor | ***impSs*** |
| impedance | bodyCompositionAnalyser | ***impee*** |
| jobModes | airConJobMode, airPurifierJobMode, clothesWasherDryerJobMode, clothesWasherJobMode, cookerHoodJobMode, dehumidifierJobMode, dishWasherJobMode, robotCleanerJobMode steamClosetJobMode, | ***jobMs*** |
| jobStates | runState | ***jobSs*** |
| kcal | bioElectricalImpedanceAnalysis | ***kcal*** |
| keyNumber | keypad | ***keyNr*** |
| latitude | geoLocation | ***latie*** |
| level | battery | ***lvl*** |
| light | alarmSpeaker | ***light*** |
| liquidLevel | liquidLevel | ***liqLl*** |
| liquidRemaining | liquidRemaining | ***liqRg*** |
| locale | location | ***locae*** |
| lock | lock | ***lock*** |
| loginName | credentials | ***logNe*** |
| longitude | geoLocation | ***longe*** |
| loudness | acousticSensor | ***louds*** |
| lowBattery | battery | ***lowBy*** |
| lqi | signalStrength | ***lqi*** |
| machineStates | runState | ***macSs*** |
| manufacturer | dmDeviceInfo | ***manur*** |
| manufacturerDetailsLink | dmDeviceInfo | ***maDLk*** |
| manufacturingDate | dmDeviceInfo | ***manDe*** |
| material | battery | ***matel*** |
| maxHeatingLevel | heatingZone | ***maHLl*** |
| maxLength | textMessage | ***maxLh*** |
| maxLevel | openLevel | ***maxLl*** |
| maxSpeed | airFlow | ***maxSd*** |
| maxValue | audioVolume, numberValue, ozoneMeter, temperature | ***maxVe*** |
| meanPressure | sphygmomanometer | ***meaPe*** |
| measuringScope | energyConsumption | ***meaSe*** |
| mediaID | mediaSelect | ***medID*** |
| mediaName | mediaSelect | ***medNe*** |
| mediaType | mediaSelect | ***medTe*** |
| memorySize | 3Dprinter | ***memSe*** |
| messageEncoding | textMessage | ***mesEg*** |
| metadata | features | ***metaa*** |
| minLength | textMessage | ***minLh*** |
| minLevel | openLevel | ***minLl*** |
| minSpeed | airFlow | ***minSd*** |
| minValue | numberValue, temperature | ***minVe*** |
| modality | pulsemeter | ***moday*** |
| model | dmDeviceInfo | ***model*** |
| monitoringEnabled | airQualitySensor | ***monEd*** |
| multiFirmware | dmFirmware | ***mulFe*** |
| multiplyingFactors | energyConsumption, energyGeneration | ***mulFs*** |
| muscle | bioElectricalImpedanceAnalysis | ***musce*** |
| muscleMass | bodyCompositionAnalyser | ***musMs*** |
| muteEnabled | audioVolume | ***mutEd*** |
| name | dmPackage, dmSoftware | ***name*** |
| network | 3Dprinter | ***netwk*** |
| numberValue | numberValue | ***numVe*** |
| object | binaryObject | ***objet*** |
| objectType | binaryObject | ***objTe*** |
| openAlarm | doorStatus | ***opeAm*** |
| openDuration | doorStatus | ***opeDn*** |
| openLevel | openLevel | ***opeLl*** |
| originID | origin | ***oriID*** |
| osVersion | dmDeviceInfo | ***oseVn*** |
| overcurrentStatus | overcurrentSensor | ***oveSs*** |
| oxygenSaturation | oximeter | ***oxySn*** |
| ozoneStatus | ozoneMeter | ***ozoSs*** |
| ozoneValueMG | ozoneMeter | ***ozVMG*** |
| ozoneValuePPM | ozoneMeter | ***oVPPM*** |
| password | credentials | ***pwd*** |
| postalAddress | location | ***posAs*** |
| power | energyConsumption | ***power*** |
| powerGenerationData | energyGeneration | ***poGDa*** |
| powerSaveEnabled | powerSave | ***poSEd*** |
| powerState | binarySwitch | ***powSe*** |
| powerStatus | dmAgent | ***powSs*** |
| precision | features | ***precn*** |
| presentationURL | dmDeviceInfo | ***prURL*** |
| previousChannel | televisionChannel | ***preCl*** |
| preWash | clothesWasherJobModeOption | ***preWh*** |
| primaryName | dmFirmware | ***priNe*** |
| primaryState | dmFirmware | ***priSe*** |
| primaryUrl | dmFirmware | ***priUl*** |
| primaryVersion | dmFirmware | ***priVn*** |
| printingState | printQueue | ***priS0*** |
| printSizeX | 3Dprinter | ***priSX*** |
| printSizeY | 3Dprinter | ***priSY*** |
| printSizeZ | 3Dprinter | ***priSZ*** |
| printType | 3Dprinter | ***priTe*** |
| progressPercentage | runState | ***proPe*** |
| pulseRate | pulsemeter | ***pulRe*** |
| pushed | pushButton | ***pushd*** |
| ramAvailable | dmAgent | ***ramAe*** |
| ramTotal | dmAgent | ***ramTl*** |
| rapidCool | refrigeration | ***rapCl*** |
| rapidFreeze | refrigeration | ***rapFe*** |
| recipientID | phoneCall | ***recID*** |
| red | colour | ***red*** |
| referenceTimer | timer | ***refTr*** |
| relativeHumidity | relativeHumidity | ***relHy*** |
| reliability | features | ***reliy*** |
| remoteControlEnabled | remoteControlEnable | ***reCEd*** |
| resistance | bioElectricalImpedanceAnalysis | ***resie*** |
| room | location | ***room*** |
| roundingEnergyConsumption | energyConsumption | ***roECn*** |
| roundingEnergyGeneration | energyGeneration | ***roEGn*** |
| rr | pulsemeter | ***r0*** |
| rssi | signalStrength | ***rssi*** |
| runningTime | timer | ***runTe*** |
| sdp | sessionDescription | ***sdp*** |
| secondaryName | dmFirmware | ***secNe*** |
| secondaryState | dmFirmware | ***secSe*** |
| secondaryUrl | dmFirmware | ***secUl*** |
| secondaryVersion | dmFirmware | ***secVn*** |
| securityModes | securityMode | ***secMs*** |
| sensitivity | motionSensor | ***sensy*** |
| sensorHumidity | airQualitySensor | ***senHy*** |
| sensorOdor | airQualitySensor | ***senOr*** |
| sensorPM1 | airQualitySensor | ***sePM1*** |
| sensorPM10 | airQualitySensor | ***sePM0*** |
| serialNumber | dmDeviceInfo | ***serNr*** |
| sensorPM2 | airQualitySensor | ***sePM2*** |
| significantDigits | energyConsumption, energyGeneration | ***sigDs*** |
| silentTime | motionSensor | ***silTe*** |
| size | binaryObject, features | ***size*** |
| smokeThreshhold | smokeSensor | ***smoTd*** |
| softLeanMass | bodyCompositionAnalyser | ***soLMs*** |
| soilLevel | clothesWasherJobModeOption | ***soiLl*** |
| speed | airFlow | ***speed*** |
| speedFactor | playerControl | ***speFr*** |
| speedWash | clothesWasherJobModeOption | ***speWh*** |
| spinLevelStrength | spinLevel | ***spLSh*** |
| spinSpeed | clothesWasherJobModeOption | ***spiSd*** |
| startPause | operationMode | ***staPe*** |
| state | dmAgent, dmPackage, dmSoftware | ***state*** |
| status | boiler, dmEventLog, electricVehicleConnector, faultDetection, filterInf, mediaSelect | ***sus*** |
| steamTreat | clothesWasherJobModeOption | ***steTt*** |
| step | numberValue | ***step*** |
| stepValue | audioVolume, openLevel, temperature | ***steVe*** |
| storageAvailable | dmAgent | ***stoAe*** |
| storageTotal | dmAgent | ***stoTl*** |
| strength | brewing | ***streh*** |
| subModel | dmDeviceInfo | ***subMl*** |
| supportedHorizontalDirection | airFlow | ***suHDn*** |
| supportedMediaSources | mediaSelect | ***suMSs*** |
| supportedMessageValues | textMessage | ***suMVs*** |
| supportedPlayerModes | playerControl | ***suPMs*** |
| supportedVerticalDirection | airFlow | ***suVDn*** |
| supportURL | dmDeviceInfo | ***suURL*** |
| swVersion | dmDeviceInfo | ***sweVn*** |
| systemTime | dmAgent | ***sysTe*** |
| systolicPressure | sphygmomanometer | ***sysPe*** |
| targetAltitude | geoLocation | ***tarAe*** |
| targetDuration | timer | ***tarDn*** |
| targetLatitude | geoLocation | ***tarLe*** |
| targetLongitude | geoLocation | ***tarL0*** |
| targetTemperature | temperature | ***tarTe*** |
| targetTimeToStart | timer | ***tTTSt*** |
| targetTimeToStop | timer | ***tTTSp*** |
| temperature | temperatureAlarm | ***tempe*** |
| temperatureThreshhold | temperatureAlarm | ***temTd*** |
| textMessage | textMessage | ***texMe*** |
| time | keepWarm | ***time*** |
| token | credentials | ***tk*** |
| tone | alarmSpeaker | ***tone*** |
| turboEnabled | turbo | ***turEd*** |
| type | dmEventLog | ***type*** |
| unit | temperature | ***unit*** |
| uri | printQueue | ***ur0*** |
| url | sessionDescription, dmPackage, dmSoftware | ***url*** |
| useGrinder | grinder | ***useGr*** |
| uvStatus | uvSensor | ***uvtSs*** |
| uvValue | uvSensor | ***uvaVe*** |
| version | dmPackage, dmSoftware | ***versn*** |
| verticalAccuracy | geoLocation | ***verAy*** |
| verticalDirection | airFlow | ***verDn*** |
| visceraFat | bioElectricalImpedanceAnalysis | ***visFt*** |
| voc | airQualitySensor | ***voc*** |
| voltage | battery | ***volte*** |
| voltage | energyConsumption | ***volte*** |
| volumePercentage | audioVolume | ***volPe*** |
| washTemp | clothesWasherJobModeOption | ***wasTp*** |
| water | bioElectricalImpedanceAnalysis | ***water*** |
| waterFlowStrength | waterFlow | ***waFSh*** |
| weight | weight | ***weigt*** |

In protocol bindings resource attributes names for properties of sub-devices shall be translated into short names of Table 6.3.3-3.

Table 6.3.3-3: Resource attribute short names (SubDevice properties)

|  |  |  |
| --- | --- | --- |
| Attribute Name | Occurs in | Short Name |
| propAreaNwkType | dmAreaNwkInfo | ***pANTe*** |

### Resource attributes for actions arguments

In protocol bindings resource attributes names for arguments of actions shall be translated into short names of Table 6.3.4-1.

Table 6.3.4‑1: Resource attribute short names (Action arguments)

|  |  |  |
| --- | --- | --- |
| Argument Name | Occurs in | Short Name |
| address | readIO, writeIO | ***addrs*** |
| name | deployPackage | ***name*** |
| payload | writeIO | ***payld*** |
| rebootType | reboot | ***rebTe*** |
| url | deployPackage, updateFirmware, update | ***url*** |
| version | deployPackage, updateFirmware, update | ***versn*** |

## containerDefinition values

### Introduction

Each specialization has a containerDefinition attribute which can be used as a unique identifier and contains the information of the resource. In this clause, the detailed values of containerDefinition attributes in every specializations for the harmonized information model are given.

The full list of domains is:

“agriculture”, “city”, “common”, “health”, “home”, “industry”, “management”, “metadata”, “publicsafety”, “railway” and “vehicular”.

### Device models

Depending on the domain, the containerDefinition attribute of specializations for device models shall have the values that comply with the following rule.

* Rule: “org.onem2m.[domain].device.[device name]”, where [domain] is one of the domain names defined in 6.4.1. The name is chosen according to the domain in which the device is defined.

For example, the containerDefinition attribute of the specialization for the “deviceAirConditioner” device of the “home” domain shall be “org.onem2m.home.device.deviceAirConditioner”.

### ModuleClasses

Depending on the domain, the containerDefinition attribute of specializations for module classes shall have the values that comply with the following rule.

* Rule: “org.onem2m.[domain].moduleclass.[moduleclass name]”, where [domain] is one of the domain names defined in 6.4.1. The name is chosen according to the domain in which the module class is defined.

For example, the containerDefinition attribute of the specialization for the “alarmSpeaker” module class of the “common” domain shall be “org.onem2m.common.moduleclass.alarmSpeaker”, the containerDefinition attribute of the specialization for the “dmAgent” module class of the “management” domain shall be “org.onem2m.management.moduleclass.dmAgent”.

### Actions

Depending on the domain, the containerDefinition attribute of specializations for actions shall have the values

that comply with the following rule.

* Rule: “org.onem2m.[domain].action.[action name]”, where [domain] is one of the domain names defined in 6.4.1. The name is chosen according to the domain in which the action is defined.

For example, the containerDefinition attribute of the specialization for “activateClockTimer” action in the “timer” module class of the “common” domain shall be “org.onem2m.common.action.activateClocktimer”, the containerDefinition attribute of the specialization for the “activate” action of the “dmSoftware” module class of the “management” domain shall be “org.onem2m.management.action.activate”.

### 6.4.5 SubDevices

Depending on the domain, the containerDefinition attribute of specializations for sub-devices shall have the values that comply with the following rule.

- Rule: “org.onem2m.[domain].subdevice.[subDevice name]”, where [domain] is one of the following names: “agriculture”, “city”, “common”, “health”, “home”, “industry”, “railway”, “vehicular” and “management”. The name is chosen according to the domain in which the sub-device is defined.

For example, the containerDefinition attribute of specialization for “subDevicePowerOutlet” of the “common” domain shall be “org.onem2m.common.subdevice.subDevicePowerOutlet”,the containerDefinition attribute of the specialization forthe “dmAreaNwkInfo” of the “management” domain shall be “org.onem2m. management.subdevice.dmAreaNwkInfo”.

## XSD definitions

### Introduction

The present clause specifies how to name the files which define data types in XSD for Device and SubDevice models, ModuleClasss, Actions and enumerated types.

Seven SDT domains correspond to different vertical, economic domains (*Agriculture, Smart City, Health, Home, Industry, Public Safety, Railway, Vehicular*), they contain devices and modules that are specific to these domains.

*Management* domain contains transversal, Device Management modules, *Metadata* domain contains transversal, meta-information modules, *Horizontal* is only for enumerated types and *Common* is the domain that gathers devices and modules that do not pertain to a specific domain but are re-usable anywhere.

The following table defines the short names for XML name spaces and file name prefix:

|  |  |  |  |
| --- | --- | --- | --- |
| **Domain** | **XML Name Space** | **Domain Prefix** | **URI** |
| Agriculture | xmlns:agd | AGD | http://www.onem2m.org/xml/protocols/agriculturedomain |
| City | xmlns:cid | CID | http://www.onem2m.org/xml/protocols/citydomain |
| Common | xmlns:cod | COD | http://www.onem2m.org/xml/protocols/commondomain |
| Health | xmlns:hed | HED | http://www.onem2m.org/xml/protocols/healthdomain |
| Home | xmlns:hod | HOD | http://www.onem2m.org/xml/protocols/homedomain |
| Horizontal | xmlns:hd | HD | http://www.onem2m.org/xml/protocols/horizontaldomain |
| Industry | xmlns:ind | IND | http://www.onem2m.org/xml/protocols/industrydomain |
| Management | xmlns:mad | MAD | http://www.onem2m.org/xml/protocols/managementdomain |
| Metadata | xmlns:mdd | MDD | http://www.onem2m.org/xml/protocols/metadatadomain |
| PublicSafety | xmlns:psd | PSD | http://www.onem2m.org/xml/protocols/publicsafetydomain |
| Railway | xmlns:rad | RAD | http://www.onem2m.org/xml/protocols/railwaydomain |
| Vehicular | xmlns:ved | VED | http://www.onem2m.org/xml/protocols/vehiculardomain |

### XSD definitions for Device models

The XSD definitions for Device models are specified upon the following rule.

* Rule: [Domain Prefix]-[device name]-v<TS-version>.xsd where the string '<TS-version>' shall be interpreted as the version of the present document

For example, the XSD definition for deviceAirConditioner specified in TS-0023 v4.3.0 shall be “HOD-deviceAirConditioner-v4\_3\_0.xsd”

### XSD definitions for ModuleClass

The XSD definitions for ModuleClass are specified upon the following rule.

* Rule: [Domain Prefix]-mod-[ModuleClass name]-v<TS-version>.xsd where the string '<TS-version>' shall be interpreted as the version of the present document

For example, the XSD definition for alarmSpeaker specified in TS-0023 v4.3.0 shall be “COD-mod-alarmSpeaker-v4\_3\_0.xsd”

### XSD definitions for Action

The XSD definitions for Actions are specified upon the following rule.

* Rule: [Domain Prefix]-act-[action name]-v<TS-version>.xsd where the string '<TS-version>' shall be interpreted as the version of the present document

For example, the XSD definition for activateClockTimer specified in TS-0023 v4.3.0 shall be “HOD-act- activateClockTimer -v4\_3\_0.xsd”.

### XSD definitions for SubDevices

The XSD definitions for SubDeices are specified upon the following rule.

* Rule: [Domain Prefix]-[SubDevice name]-v<TS-version>.xsd where the string '<TS-version>' shall be interpreted as the version of the present document.

For example, the XSD definition for subDeviceCuff specified in TS-0023 v4.3.0 shall be “COD-subDeviceCuff-v4\_3\_0.xsd”.

### 6.5.6 XSD definitions for Enumerated Types

The XSD definitions for enumerated types are specified upon the following rule.

* Rule: HD-enumerationTypes-v<TS-version>.xsd where the string '<TS-version>' shall be interpreted as the version of the present document.

This file contains the definitions of all enumerated types, and nothing else.

# Mapping with Other Information Models from External Organizations

This clause specifies how the Home Appliance Information Model (HAIM) defined in the clause 5of the present document can be mapped with existing external models from , OCF, ECHONET, OMA GotAPI etc. and introduction of these models is written in annex B. The mapping shall be to enable the interworking between the oneM2M system and external technologies at the information model level. This means a oneM2M native application which understand only oneM2M standardized HAIM shall be able to interact with non-oneM2M home appliances of different technologies in a consistent way without knowing the technology specific details. An IPE shall be responsible for translating the HAIM to/from technology specific information model bidirectionally following the mapping specification in this clause. Using HAIM as a bridge, home appliances and applications of different technologies shall be able to also interact with each other via the oneM2M system (with IPEs).

## OMA GotAPI(DWAPI)

### Introduction

The following clauses are intended to specify the mapping relationship between HAIM and OMA DWAPI with tables.

OMA DWAPI (Device Web Application Programming Interface) [7] is based on OMA GotAPI (Generic Open Terminal Application Programming Interface) [19] and supports Personal Healthcare Devices (DWAPI-PCH) and 3D printer (DWAPI-3DP).

OMA DWAPI has no concept that corresponds to ModuleClass in oneM2M. The mappings of DataPoints to data objects of OMA DWAPI are expressed in following clauses.

### Device Models

#### device3Dprinter

The device3Dprinter of HAIM shall be mapped to 3D printer of OMA DWAPI-3DP on the basis of the following table.

Table 7.1.2‑1: Map of device3Dprinter of oneM2M HAIM to OMA DWAPI-3DP

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ModuleClass** | **Data Points of**  **oneM2M HAIM** | **data objects of**  **OMA DWAPI-3DP** |  | **Description** |
| binarySwitch | powerState | - |  | See clause 5.3.1.12.  The powerState is not supporteded in OMA DWAPI-3DP data object. The power state is assumed power-on in OMA DWAPI-3DP. |
| faultDetection | code | operatingStatus |  | See clause 5.3.1.34.  It shall be the integer type at HAIM, but shall be the string type at OMA DWAPI-3DP. See the Table 7.1.2‑2 . |
| 3Dprinter | printType | printType |  | See clause 5.3.1.1. |
| printSizeX | printSizeX |  |
| printSizeY | printSizeY |  |
| printSizeZ | printSizeZ |  |
| network | network |  |
| memorySize | memorySize |  |
| runState | currentMachineState | operatingStatus |  | See clause 5.3.1.75.  This value represents the machineState of the 3D printer itself. This value SHALL be interpreted by using hd:enumMachineState and generated operatingStatus as a string.  See the Table 7.1.2‑2 . |
| temperature | currentTemperature | nozzleTemp |  | See clause 5.3.1.87.  This value represents the temperature of the nozzle. This value SHALL be a float number in a range from 0.0 to 1000.0.  The unit is C. |
| printQueue | uri | uri |  | See clause 5.3.1.67. |
| printingState | msg |  | See clause 5.3.1.67.  This value represents the machineState of the queued printing job. This value shall be interpreted by using hd:enumMachineState and generated msg as a string.  See the Table 7.1.2‑3. |

Table 7.1.2‑2: Map of hd:enumMachineState of oneM2M HAIM to operatingStatus of OMA DWAPI-3DP

|  |  |  |  |
| --- | --- | --- | --- |
| **Value** | **Interpretation** | **operatingStatus of OMA DWAPI-3DP** | **Note** |
| 1 | idle | RDY | Ready to use |
| 2 | preActive |  | Not available |
| 3 | active | RUN | Under printing operation |
| 4 | reserved |  | Not available |
| 5 | stopped |  | Not available |
| 6 | error | MAN | Maintenance needed |
| 7 | diagnostic |  | Not available |
| 8 | test |  | Not available |
| 9 | maintenance | MAN | Maintenance needed |
| 10 | clear | CLR | Printing completed but the result is not removed yet |
| 11 | charging |  | Not available |

Table 7.1.2‑3: Map of hd:enumJobState of oneM2M HAIM to msg of OMA DWAPI-3DP

|  |  |  |  |
| --- | --- | --- | --- |
| **Value** | **Interpretation** | **msg of OMA DWAPI-3DP** | **Note** |
| 1 | aborted |  | Not available |
| 2 | cancelled |  | Not available |
| 3 | completed | Completed |  |
| 4 | paused | Waiting |  |
| 5 | pending | Waiting |  |
| 6 | processing | Good Start |  |

#### deviceBloodPressureMonitor

DeviceBloodPressureMonitoer of HAIM shall be mapped to Blood Pressure Monitor of OMA DWAPI-PCH on the basis of the following table.

Table 7.1.2‑4: Map of deviceBloodPressureMonitor of oneM2M HAIM to OMA DWAPI-PCH

|  |  |  |  |
| --- | --- | --- | --- |
| ModuleClass | DataPoints of  oneM2M HAIM | data objects of  OMA DWAPI-PCH | Description |
| sphygmomanometer | diastolicPressure | diastolic | See clause 5.3.1.83. |
| systolicPressure | systolic |
| meanPressure | mean |
| pulsemeter | pulseRate | pulse | See clause 5.3.1.68 |
| battery | level | batteryLevel | See clause 5.3.1.10.  It shall be the integer type at HAIM, but shall be the float type at OMA DWAPI-PCH.  It shall be the rounded percentage of the current level of battery in the range of [0, 100] at HAIM, but shall be a float number in the range of [0.0, 1.0] atOMA DWAPI-PCH. |

#### deviceGlucosemeter

DeviceGlucometer of HAIM shall be mapped to Glucometer of OMA DWAPI-PCH on the basis of the following table.

Table 7.1.2‑5: Map of deviceGlucometer of oneM2M HAIM to OMA DWAPI-PCH

|  |  |  |  |
| --- | --- | --- | --- |
| ModuleClass | DataPoints of  oneM2M HAIM | data objects of  OMA DWAPI-PCH | Description |
| glucometer | concentration | concentration | See clause 5.3.1.42. |
| hba1c | hba1c |
| contextExercise | contextExercise |
| contextMedication | contextMedication |
| contextCarbohydratesAmount | contextCarbohydrates |
| contextCarbohydratesSource | contextCarbohydrates |
| contextMeal | contextMeal |
| contextLocation | contextLocation |
| contextTester | contextTester |
| contextHealth | contextHealth |
| battery | level | batteryLevel | See clause 5.3.1.10.  It shall be the integer type at HAIM, but shall be the float type at OMA DWAPI-PCH.  It shall be the rounded percentage of the current level of battery in the range of [0, 100] at HAIM, but shall be a float number in the range of [0.0, 1.0] atOMA DWAPI-PCH. |

#### devicePulseOximeter

DevicePulseOximeter of HAIM shall be mapped to Pulse Oximeter of OMA DWAPI-PCH on the basis of the following table.

Table 7.1.2‑6: Map of devicePulseOximeter of oneM2M HAIM to OMA DWAPI-PCH

|  |  |  |  |
| --- | --- | --- | --- |
| ModuleClass | DataPoints of  oneM2M HAIM | data objects of  OMA DWAPI-PCH | Description |
| pulsemeter | modality | spo2, pulse | See clause 5.3.1.68.  When oximeter module doesn’t exist, pulsemeter module is mandatory. |
| oximeter | modality | spo2, pulse | See clause 5.3.1.59.  When pulsemeter module doesn’t exist, oximeter module is mandatory. |
| battery | level | batteryLevel | See clause 5.3.1.10.  It shall be the integer type at HAIM, but shall be the float type at OMA DWAPI-PCH.  It shall be the rounded percentage of the current level of battery in the range of [0, 100] at HAIM, but shall be a float number in the range of [0.0, 1.0] atOMA DWAPI-PCH. |

#### deviceThermometer

DeviceTermometer of HAIM shall be mapped to Thermometer of OMA DWAPI-PCH on the basis of the following table.

Table 7.1.2‑7: Map of deviceThermometer of oneM2M HAIM to OMA DWAPI-PCH

|  |  |  |  |
| --- | --- | --- | --- |
| ModuleClass | Data Points of  oneM2M HAIM | data objects of  OMA DWAPI-PCH | Description |
| temperature | unit | temperature | See clause 5.3.1.87. |
| battery | level | batteryLevel | See clause 5.3.1.10.  It shall be the integer type at HAIM, but shall be the float type at OMA DWAPI-PCH.  It shall be the rounded percentage of the current level of battery in the range of [0, 100] at HAIM, but shall be a float number in the range of [0.0, 1.0] atOMA DWAPI-PCH. |

#### deviceWeightScaleAndBodyCompositionAnalyser

DeviceWeightScaleAdBodyCompositionAnalyser of HAIM shall be mapped to Weight Scale Body Composition Analyser of OMA DWAPI-PCH on the basis of the following table.

Table 7.1.2‑8: Map of deviceWeightScaleAdBodyCompositionAnalyser of oneM2M HAIM to OMA DWAPI-PCH

|  |  |  |  |
| --- | --- | --- | --- |
| ModuleClass | DataPoints of  oneM2M HAIM | data objects of  OMA DWAPI-PCH | Description |
| bodyCompositionAnalyser | bodyLength | bodyLength | See clause 5.3.1.14. |
| Bmi | bmi |
| fatFreeMass | fatFreeMass |
| softLeanMass | softLeanMass |
| muscleMass | muscleMass |
| basalMetabolism | basalMetabolism |
| impedance | impedance |
| weight | weight | bodyMass | See clause 5.3.1.99. |
| bioElectricalImpedanceAnalysis | water | bodyWater | See clause 5.3.1.13. |
| fat | bodyFat |
| muscle | musclePercentage |
| battery | level | batteryLevel | See clause 5.3.1.10.  It shall be the integer type at HAIM, but shall be the float type at OMA DWAPI-PCH.  It shall be the rounded percentage of the current level of battery in the range of [0, 100] at HAIM, but shall be a float number in the range of [0.0, 1.0] atOMA DWAPI-PCH. |

#### deviceHeartRateMonitor

DeviceHeartRateMonitor of HAIM shall be mapped to Heart Rate Monitor of OMA DWAPI-PCH on the basis of the following table.

Table 7.1.2‑9: Map of deviceHeartRateMonitor of oneM2M HAIM to OMA DWAPI-PCH

|  |  |  |  |
| --- | --- | --- | --- |
| ModuleClass | DataPoints of  oneM2M HAIM | data objects of  OMA DWAPI-PCH | Description |
| pulsemeter | pulseRate | rate | See clause 5.3.1.68. |
| rr | rr |
| energy | energy |
| battery | level | batteryLevel | See clause 5.3.1.10.  It shall be the integer type at HAIM, but shall be the float type at OMA DWAPI-PCH.  It shall be the rounded percentage of the current level of battery in the range of [0, 100] at HAIM, but shall be a float number in the range of [0.0, 1.0] atOMA DWAPI-PCH. |

### Data Types

Data types of oneM2M HAIM and OMA DWAPI-PCH shall be mapped each other on the basis of the following table.

Table 7.1.3‑1: Map of data types between oneM2M and OMA DWAPI-PCH

|  |  |  |
| --- | --- | --- |
| oneM2M data type | Mapping to data type in OMA DWAPI | Description |
| xs:integer | int, number, string | Data type for 32-bit signed integer.  For indicating 3D printerState, the integer value should be interpeted into string by referring the tables in clause 7.1.2.1 . |
| xs:string | string, array | Data type for text. The length limitation should be considered for the mapping. |
| xs:float | float | Data type for a single precision 32-bit floating point type as defined in XML Schema 1.0 [14] as the float primitive type. |
| xs:boolean | boolean | Data type for Boolean. |

# Ontology for the Home Appliance Information Model aligned with oneM2M Base Ontology

The following table shows a mapping of the Home Appliance Information Model to the oneM2M Base Ontology in oneM2M TS-0012 [i.5].

The table only shows mapping of SDT concepts that are used to classify all concepts in the Home Appliance Information Model. Therefore, since any concept in the Home Appliance Information Model can be classified according to a specific SDT concept it also (transitively) maps to the related class of the oneM2M Base Ontology.

Table 7.1.3‑1: Mapping between SDT concepts in the Home Appliance Information Model   
and the oneM2M Base Ontology

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SDT Concept in the Home Appliance Information Model | Mapping relationship | Class in Base Ontology | Property in Base Ontology | Comment |
| SDT: Device | sub-class of | Device |  |  |
| SDT: SubDevice | sub-class of | Device |  | The base ontology allows a Device to consist of (sub-) Devices |
| SDT: Action | sub-class of | Operation |  |  |
| SDT: Args (of an Action) | sub-class of | OperationInput |  |  |
| SDT: ReturnType (of an Action) | sub-class of | OperationOutput |  |  |
| SDT: Event | sub-class of | Operation |  |  |
| SDT: Data (of an Event) | sub-class of | OutputDataPoint |  |  |
| SDT: Module | sub-class of | Service |  | The base ontology allows a Service to have subServices. Each SDT:Module implements one SDT:ModuleClass.  Therfore SDT:Module can be considered a subclass of SDT:ModuleClass and therefore subclass of oneM2M:Service.  See note. |
| SDT: ModuleClass | sub-class of | Service |  | See note |
| SDT: UnitOfMeasure | sub-class of | MetaData |  |  |
| SDT: DataPoint | sub-class of | InputDataPoint |  | If SDT:DataPoint is writable |
| SDT: DataPoint | sub-class of | OutputDataPoint |  | If SDT:DataPoint is readable |
| SDT: Property (of a Device) | sub-class of | ThingProperty |  |  |
| SDT: Property (of a ModuleClass) | sub-class of | Aspect |  | Aspect (of the Functionality) |
| SDT: SimpleType | sub-property of |  | hasDataType | The base ontology's SimpleTypeVariable class has data properties:   * hasDataType * hasDataRestriction |
| SDT: Constraint | sub-property of |  | hasDataRestriction |  |
| NOTE: In RESTful technologies the Service (i.e. the electronic representation of a Functionality in a network) is implicitly bound to its Functionality by the naming of the used resources (e.g. the Functionality of ModuleClass "AudioVolume" is implemented as a Service through CRUD operations on a [audioVolume] <flexContainer> specialization). | | | | |

1. (informative):   
   Resource Mapping Examples
   1. Introduction

The AE may construct oneM2M resource tree on hosting CSE as the mapping of associated device, and each XSD definition for the device information models is generated following ‘Resource Mapping Rule' in clause 6.2.

The present clause explains how to use the oneM2M resource tree to map Device model for each device (see clause 5.5).

* 1. Example for Device model ‘deviceAirConditioner'

The present clause explains the creation process for the device typed 'deviceAirConditioner' (see clause 5.5.1.1 for device model definition of ‘deviceAirConditioner').

Using the definition, 'deviceAirConditioner' model is mapped to [deviceAirConditioner] resource which is a specialization of <flexContainer> resource (See Figure A.2-1).

Figure A.2-1: Structure of *[deviceAirConditioner]* resource

The AE creates the [deviceAirConditioner] specialization of <flexContainer> resource for the Device model [deviceAirConditioner] resource.

The [deviceAirConditioner] resource contains the child resource specified in Table A.2-2.

Table A.2-2: Child resources of *[deviceAirConditioner]* resource

| Child Resources of *[deviceAirConditioner]* | Child Resource Type | Multiplicity | Description |
| --- | --- | --- | --- |
| *[variable]* | *<flexContainer> as defined in the specialization [binarySwitch]* | 0..1 | This resource is used to map 'binarySwith' ModuleClass defined in clause 5.3.1.12. |
| *[variable]* | *<flexContainer> as defined in the specialization [runState]* | 0..1 | This resource is used to map 'runState' ModuleClass defined in clause 5.3.1.75. |
| *[variable]* | *<flexContainer> as defined in the specialization [airConJobMode]* | 0..1 | This resource is used to map ‘airConJobMode’ ModuleClass defined in clause.  Editor’s Note: airConJobMode is not a moduleclass. It is an instance of that. It is needed to fix. |
| *[variable]* | *<flexContainer> as defined in the specialization [airConOperationMode]* | 0..1 | This resource is used to map ‘airConOperationMode’ ModuleClass defined in clause 5.3.1.57. |
| *[variable]* | *<flexContainer> as defined in the specialization [airCleanOperationMode]* | 0..1 | This resource is used to map ‘airCleanOperationMode’ ModuleClass defined in clause 5.3.1.57. |
| *[variable]* | *<flexContainer> as defined in the specialization [temperature]* | 0..1 | This resource is used to map ‘temperature’ ModuleClass defined in clause 5.3.1.87. |
| *[variable]* | *<flexContainer> as defined in the specialization [timer]* | 0..1 | This resource is used to map 'timer' ModuleClass defined in clause 5.3.1.90. |
| *[variable]* | *<flexContainer> as defined in the specialization [sleepTimer]* | 0..1 | This resource is used to map 'sleepTimer' ModuleClass defined in clause 5.3.1.90. |
| *[variable]* | *<flexContainer> as defined in the specialization [turbo]* | 0..1 | This resource is used to map 'turbo' ModuleClass defined in clause 5.3.1.91. |
| *[variable]* | *<flexContainer> as defined in the specialization [airFlow]* | 0..1 | This resource is used to map 'airFlow' ModuleClass defined in clause 5.3.1.4. |
| *[variable]* | *<flexContainer> as defined in the specialization [powerSave]* | 0..1 | This resource is used to map 'powerSave' ModuleClass defined in clause 5.3.1.66. |
| *[variable]* | *<flexContainer> as defined in the specialization [airQualitySensor]* | 0..1 | This resource is used to map 'airQualitySensor' ModuleClass defined in clause 5.3.1.6. |
| *[variable]* | *<flexContainer> as defined in the specialization [filterInfo]* | 0..1 | This resource is used to map 'filterInfo' ModuleClass defined in clause 5.3.1.35. |
| *[variable]* | *<subscription>* | 0..n | See clause 9.6.8 in oneM2M TS-0001 [i.3] |

Editor’s Note: Above table should be updated compliant to present structure of deviceAirConditioner.

The [deviceAirConditioner] resource contains the attributes specified in Table A.2-3.

Table A.2-3: Attributes of *[deviceAirConditioner]* resource

| Attributes of  *[deviceAirConditioner]* | Multiplicity | RW/  RO/  WO | Description |
| --- | --- | --- | --- |
| *resourceType* | 1 | RO | See clause 9.6.1.3 in oneM2M TS-0001 [i.3]. |
| *resourceID* | 1 | RO | See clause 9.6.1.3 in oneM2M TS-0001 [i.3]. |
| *resourceName* | 1 | RO | See clause 9.6.1.3 in oneM2M TS-0001 [i.3]. |
| *parentID* | 1 | RO | See clause 9.6.1.3 in oneM2M TS-0001 [i.3]. |
| *expirationTime* | 1 | RW | See clause 9.6.1.3 in oneM2M TS-0001 [i.3]. |
| *accessControlPolicyIDs* | 0..1 (L) | RW | See clause 9.6.1.3 in oneM2M TS-0001 [i.3]. |
| *creationTime* | 1 | RO | See clause 9.6.1.3 in oneM2M TS-0001 [i.3]. |
| *lastModifiedTime* | 1 | RO | See clause 9.6.1.3 in oneM2M TS-0001 [i.3]. |
| *labels* | 0..1 | RW | See clause 9.6.1.3 in oneM2M TS-0001 [i.3]. |
| *dynamicAuthorizationConsultationIDs* | 0..1 (L) | RW | See clause 9.6.1.3 in oneM2M TS-0001 [i.3] |
| *stateTag* | 1 | RO | See clause 9.6.1.3 in oneM2M TS-0001 [i.3] |
| *creator* | 0..1 | RW | See clause 9.6.35 in oneM2M TS-0001 [i.3] |
| *containerDefinition* | 1 | WO | The value is "org.onem2m.home.device.airconditioner" |
| *ontologyRef* | 0..1 | RW | See clause 9.6.35 in oneM2M TS-0001 [i.3]. |
| *contentSize* | 1 | RO | See clause 9.6.35 in oneM2M TS-0001 [i.3]. |
| *nodeLink* | 0..1 | RO | nodeLink attribute links to a <node> resource that is hosted on the same hosting CSE of the <flexContainer>. See clause 6.2.2 and 6.2.5 for more details. |

* 1. Example of ModuleClass 'binarySwitch'

The [*binarySwitch*] resource is used to share information regarding the modeled binary switch module as a ModuleClass. The [*binarySwitch*] resource is a specialization of the <*flexContainer*> resource.



Figure A.3-1: Structure of *[binarySwitch]* resource

The *[binarySwitch]* resource contains the child resource specified in Table A.3-2.

Table A.3-2: Child resources of *[binarySwitch]* resource

| Child Resources of *[binarySwitch]* | Child Resource Type | Multiplicity | Description |
| --- | --- | --- | --- |
| *[variable]* | *<flexContainer> as defined in the specialization [toggle]* | 0..1 | This resource is used to map 'toggle' Action defined in Clause 5.3.1.12. |
| *[variable]* | *<subscription>* | 0..n | See clause 9.6.8 in oneM2M TS-0001 [i.3] |

The *[binarySwitch]* resource contains the attributes specified in Table A.3-3.

Table A.3-3: Attributes of *[binarySwitch]* resource

| Attributes of  *[binarySwitch]* | Multiplicity | RW/  RO/  WO | Description |
| --- | --- | --- | --- |
| *resourceType* | 1 | RO | See clause 9.6.1.3 in oneM2M TS-0001 [i.3] |
| *resourceID* | 1 | RO | See clause 9.6.1.3 in oneM2M TS-0001 [i.3] |
| *resourceName* | 1 | RO | See clause 9.6.1.3 in oneM2M TS-0001 [i.3] |
| *parentID* | 1 | RO | See clause 9.6.1.3 in oneM2M TS-0001 [i.3] |
| *expirationTime* | 1 | RW | See clause 9.6.1.3 in oneM2M TS-0001 [i.3] |
| *accessControlPolicyIDs* | 0..1 (L) | RW | See clause 9.6.1.3 in oneM2M TS-0001 [i.3] |
| *creationTime* | 1 | RO | See clause 9.6.1.3 in oneM2M TS-0001 [i.3] |
| *lastModifiedTime* | 1 | RO | See clause 9.6.1.3 in oneM2M TS-0001 [i.3] |
| *labels* | 0..1 | RW | See clause 9.6.1.3 in oneM2M TS-0001 [i.3] |
| *dynamicAuthorizationConsultationIDs* | 0..1 (L) | RW | See clause 9.6.1.3 in oneM2M TS-0001 [i.3] |
| *stateTag* | 1 | RO | See clause 9.6.1.3 in oneM2M TS-0001 [i.3] |
| *creator* | 0..1 | RW | See clause 9.6.35 in oneM2M TS-0001 [i.3] |
| *containerDefinition* | 1 | WO | The value is "org.onem2m.home.moduleclass.binaryswitch" |
| *ontologyRef* | 0..1 | RW | See clause 9.6.35 in oneM2M TS-0001 [i.3] |
| *contentSize* | 1 | RO | See clause 9.6.35 in oneM2M TS-0001 [i.3]. |
| *nodeLink* | 0..1 | RW | Not applicable to a ModuleClass specialization. This attribute is not present in an instantiation of this resource. |
| *dataGenerationTime* | 0..1 | RO | See clause 6.2.3 |
| *powerState* | 1 | RW | See clause 5.3.1.12 |

* 1. Example of Action 'toggle'

The [*toggle*] resource is used to share information regarding the modeled toggle as an Action. The [*toggle*] resource is a specialization of the <*flexContainer*> resource.



Figure A.4-1: Structure of *[toggle]* resource

The *[toggle]* resource contains the child resource specified in Table A.4-2.

Table A.4-2: Child resources of *[toggle]* resource

| Child Resources of *[toggle]* | Child Resource Type | Multiplicity | Description |
| --- | --- | --- | --- |
| *[variable]* | *<subscription>* | 0..n | See clause 9.6.8 in oneM2M TS-0001 [i.3] |

The *[toggle]* resource contains the attributes specified in Table A.4-3.

Table A.4-3: Attributes of *[toggle]* resource

| Attributes of  *[toggle]* | Multiplicity | RW/  RO/  WO | Description |
| --- | --- | --- | --- |
| *resourceType* | 1 | RO | See clause 9.6.1.3 in oneM2M TS-0001 [i.3] |
| *resourceID* | 1 | RO | See clause 9.6.1.3 in oneM2M TS-0001 [i.3] |
| *resourceName* | 1 | RO | See clause 9.6.1.3 in oneM2M TS-0001 [i.3] |
| *parentID* | 1 | RO | See clause 9.6.1.3 in oneM2M TS-0001 [i.3] |
| *expirationTime* | 1 | RW | See clause 9.6.1.3 in oneM2M TS-0001 [i.3] |
| *accessControlPolicyIDs* | 0..1 (L) | RW | See clause 9.6.1.3 in oneM2M TS-0001 [i.3] |
| *creationTime* | 1 | RO | See clause 9.6.1.3 in oneM2M TS-0001 [i.3] |
| *lastModifiedTime* | 1 | RO | See clause 9.6.1.3 in oneM2M TS-0001 [i.3] |
| *labels* | 0..1 | RW | See clause 9.6.1.3 in oneM2M TS-0001 [i.3] |
| *dynamicAuthorizationConsultationIDs* | 0..1 (L) | RW | See clause 9.6.1.3 in oneM2M TS-0001 [i.3] |
| *stateTag* | 1 | RO | See clause 9.6.1.3 in oneM2M TS-0001 [i.3] |
| *creator* | 0..1 | RW | See clause 9.6.35 in oneM2M TS-0001 [i.3] |
| *containerDefinition* | 1 | WO | The value is "org.onem2m.home.moduleclass.binaryswitch.toggle" |
| *ontologyRef* | 0..1 | RW | See clause 9.6.35 in oneM2M TS-0001 [i.3] |
| *contentSize* | 1 | RO | See clause 9.6.35 in oneM2M TS-0001 [i.3]. |
| *nodeLink* | 0..1 | RW | Not applicable to an Action specialization. This attribute is not present in an instantiation of this resource. |

1. (informative):   
   Introduction of External Organizations’ Data Models
   1. OMA Got API(DWAPI-PCH)

OMA GotAPI(OMA Generic Open Terminal API Framework) provides the framework to enable applications and multitype devices through GotAPI Servers and Extension Plug-Ins [6]. When APIs are implemented in Extension Plug-Ins under the GotAPI framework, these APIs are called as OMA Device WebAPIs Enabler. In case of healthcare devices, these APIs are called as OMA DWAPI-PCH(Device WebAPIs for Personal Connected Healthcare).

Healthcare devices can be a one of the smart home devices so OMA DWWAPI-PCH can have relationship with oneM2M SDT.

* 1. OCF

OCF specifications provide a common, open connectivity framework for embedded developers that enables a common device discovery and interaction model, common data model and a robust security framework whilst abstracting away the physical connectivity hardware (and related protocols).

OCF Device Specification [i.8] defines list of smart home devices. Each smart home device contains an unique identifier and list of mandatory/optional resources. Each resource definition contains an unique identifier, identification of the default interface and other supported interfaces, list of supported methods, list of allowed actions and list of the mandatory/optional property(-ies) the resource exposes.

* + 1. Introduction

This clause specifies the mapping relationship between oneM2M and OCF Devices.

* + 1. Device Type Mapping

The following table captures the equivalency mapping between OCF defined Device Types and oneM2M defined Devices. The minimum module sets for each oneM2M device is provided in this specification. The minimum resource sets for each OCF Device is provided in the OCF Device Specification [i.8].

**Table 7.1.3‑8: OCF to oneM2M Device Type Mapping**

| OCF Device Name | OCF Device Type | oneM2M Device |
| --- | --- | --- |
| Active Speaker | oic.d.speaker | N/A |
| Air Conditioner | oic.d.airconditioner | deviceAirConditioner |
| Air Purifier | oic.d.airpurifier | deviceAirPurifier |
| Air Quality Monitor | oic.d.airqualitymonitor | deviceAirQualityMonitor |
| Battery | oic.d.battery | deviceStorageBattery |
| Blind | oic.d.blind | deviceWindowShade |
| Camera | oic.d.camera | deviceCamera |
| Clothes Washer Dryer | oic.d.washerdryer | deviceClothesWasherDryer |
| Coffee Machine | oic.d.coffeemachine | deviceCoffeeMachine |
| Cooker Hood | oic.d.cookerhood | deviceCookerHood |
| Cooktop | oic.d.cooktop | deviceCooktop |
| Dehumidifier | oic.d.dehumidifier | deviceDehumidifier |
| Dishwasher | oic.d.dishwasher | deviceDishWasher |
| Door | oic.d.door | deviceDoor |
| Dryer (Laundry) | oic.d.dryer | deviceClothesDryer |
| Electric Vehicle Charger | oic.d.electricvehiclecharger | deviceElectricVehicleCharger |
| Electric Meter | oic.d.electricmeter | deviceSmartElectricMeter |
| Energy Generator | oic.d.energygenerator | deviceMicrogeneration |
| Fan | oic.d.fan | deviceFan |
| Food Probe | oic.d.foodprobe | deviceFoodProbe |
| Freezer | oic.d.freezer | deviceFreezer |
| Garage Door | oic.d.garagedoor | deviceDoor |
| Generic Sensor | oic.d.sensor | N/A |
| Grinder | oic.d.grinder | N/A |
| Humidifier | oic.d.humidifier | deviceHumidifier |
| Light | oic.d.light | deviceLight |
| Oven | oic.d.oven | deviceOven |
| Printer | oic.d.printer | devicePrinter |
| Printer Multi-Function | oic.d.multifunctionprinter | deviceMultiFunctionPrinter |
| Receiver | oic.d.receiver | deviceAudioReceiver |
| Refrigerator | oic.d.refrigerator | deviceRefrigerator |
| Robot Cleaner | oic.d.robotcleaner | deviceRobotCleaner |
| Scanner | oic.d.scanner | deviceScanner |
| Security Panel | oic.d.securitypanel | deviceSecurityPanel |
| Set Top Box | oic.d.stb | deviceSetTopBox |
| Smart Lock | oic.d.smartlock | deviceDoorLock |
| Smart Plug | oic.d.smartplug | deviceSmartPlug |
| Switch | oic.d.switch | deviceSwitch |
| Television | oic.d.tv | deviceTelevision |
| Thermostat | oic.d.thermostat | deviceThermostat |
| Washer (Laundry) | oic.d.washer | deviceClothesWasher |
| Water Heater | oic.d.waterheater | deviceWaterHeater |
| Water Valve | oic.d.watervalve | deviceWaterValve |
| Window | oic.d.window | N/A |

1. (informative):   
   Mapping to Content Attribute
   1. Introduction

Current SDT models are used only in form of <flexContainer>s, and how to design content attribute of <contentInstance> and <timeSeriesInstance> is left to developers. There is no rule for design of content attribute, it means interoperability of content attribute is low. Then SDT can become one of the rules for design of content attribute, and the low interoperability problem will be solved.

The present clause explains how to use SDT as one of the rules for design of content attribute.

There are several benefits of using SDT in content attribute.

First, the resource architecture can be more simple than the one using <flexContainer>s. When using <flexContainer>s, universal attributes are mapped either into attributes of [deviceInfo] under a <node> besides <flexContainer>s, or into custom attributes of [dmDeviceInfo] under a [flexNode] (See Rule 1-8 in clause 6.2.2). Moreover, Action Class and DataPoint Class are the same layer in SDT, but Action Class is mapped to <flexContainer> itself and DataPoint Class is mapped to attributes of <flexContainer> expressing Module class. On the other hand, Using SDT in content attribute means using only one <contentInstance> or <timeSeriesInstance> so the resource architecture is simple.

Relating this benefit, it becomes easy to understand where to write information.

Second, <contentInstance> and <timeSeriesInstance> becomes more interoperable. How to write SDT in content attribute is able to become one of designs of content attribute and the low interoperability of <contentInstance> and <timeSeriesInstance> will be solved.

Third, If useful libraries are prepared, content attribute is able to be expressed in XML/JSON/CBOR with small changes on program.

In addition, tools can generate validator of the data and converter among the supported formats

* 1. XML representation of SDT instances.

ModuleClasses, SubDevice models and DeviceClass models written in clause 5 are expressed another way with using each class names as the tag. This clause introduces this way.

Normative work for defining the mapping rules from SDT to XML/ JSON instance are defined by SDT4.0.

* + 1. Mapping Rules of XML representation

Mapping from SDT instance into XML representation is following. Inclusion relationship of SDT instances are directly expressed as inclusion relationship of XML tags. Tag name is same as each classes.

<DeviceClass Class Name>

<SubDevice Class Name>

<ModuleClass Class Name>

<DataPoint Class Name>value</DataPoint Class Name>

<Property Class Name>value</Property Class Name>

</ModuleClass Class Name>

</ SubDevice Class Name>

</DeviceClass Class Name>

Action Class can not have any value and it only lengthen the message so it is omitted.

When a certain device does not have any SubDevice, the tags about SubDevices don’t appear and tags about ModuleClass are placed under the DeviceClass directly.

* + 1. Example of XML representation

Below is the example for deviceThreeDPrinter:

<deviceThreeDPrinter>

<binarySwitch>

<powerState>True</powerState>

<toggle></toggle>

</binarySwitch>

<faultDetection>

<status>False</status>

<code></code>

<description></description>

</faultDetection>

<3Dprinter>

<printType>2</printType>

<printSizeX>70</printSizeX>

<printSizeY>80</printSizeY>

<printSizeZ>90</printSizeZ>

<network>True</network>

<memorySize>100</memorySize>

</3Dprinter>

<runState>

<currentMachineState>1</currentMachineState>

<machineStates>1,2,3</machineStates>

<currentJobState>1</currentJobState>

<jobStates>1,2,3</jobStates>

<progressPercentage>60</progressPercentage>

</runState>

<temperature>

<currentTemperature>20</currentTemperature>

<targetTemperature>23</targetTemperature>

<unit>celsius</unit>

<minValue>15</minValue>

<maxValue>28</maxValue>

<stepValue>0.1</stepValue>

</temperature>

<printQueue>

<uri> file://www.example.com/file.extension</uri>

<printingState>1</printingState>

</printQueue>

</deviceThreeDPrinter>

* 1. JSON representation of SDT instances

This clause tells how to express SDT instances with JSON.

* + 1. Mapping Rules of JSON representation

Mapping from SDT instance into JSON representation is following . Inclusion relationship of SDT instances are directly expressed as inclusion relationship of JSON hash({}). Key name is same as each class name of SDT. Value types are written in various types depending SDT definition.

{

“Device Class Name”: {

“SubDevice Class Name”: {

”Module Class Name”: {

“DataPointClassName”: value ( by specified types in SDT)

}

}

}

Action Class can not have any value and it only lengthen the message so it is omitted.

When a certain device does not any SubDevice Class, the tags about SubDevices Class don’t appare and tags about Module Class are placed under the Device Class directly.

* + 1. Example of JSON representation

Below is the example for deviceThreeDPrinter:

{

“deviceThreeDPrinter”: {

”binarySwitch”: {

“powerState”: true

},

“faultDetection”: {

“status”: false,

“code”: ””,

“description”: “”

},

“3Dprinter”: {

“printType”: 2,

“printSizeX”: 70,

“printSizeY”: 80,

“printSizeZ”: 90,

“network”: true,

“memorySize”: 100

},

“runState”: {

“currentMachineState”: 1,

“machineStates”: [1, 2, 3],

“currentJobState”: 1,

“jobState”: [1, 2, 3],

“progressPercentage”: 60,

},

“temperature”: {

“currentTemperature”: 20,

“targetTemperature”: 23,

“unit”: “celsius”,

“minValue”: 15,

“maxValue”: 28,

“stepValue”: 0.1

},

“printQueue”:{

“uri”: “file://www.example.com/file.extension”,

“printingState”: 1

}

}

* 1. How to write into *content* attribute

Any size of the SDT class cluster may be mapped to *content* attribute. For example, from only the DataPoint class to DataPoint, Action, Module, SubDevice and Device classes may be mapped to one *content* attribute at once.

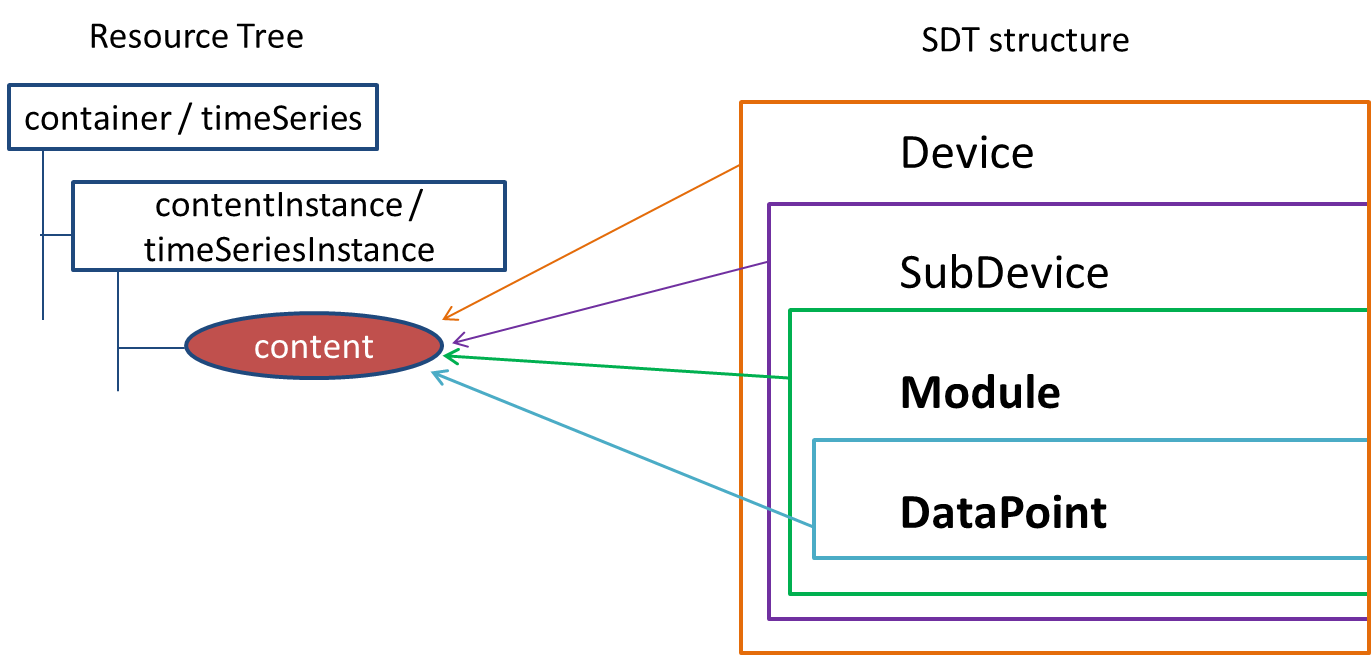


Figure C‑4: Image of Mapping SDT to *content* attribute

An example that expresses a CREATE request for <contentInstance> serialized into an XML document is shown below. This example is for the Module class and DataPoint class mapping and only *content* attribute and *contentInfo* attribute are shown as content parameter (pc).

<?xml version="1.0" encoding="UTF-8"?>

<m2m:rqp xmlns:m2m="http://www.onem2m.org/xml/protocols">

<op>1</op>

<to>//example.net/myCSE/-/Cont1</to>

<fr>/myCSE/C2345</fr>

<rqi>0002bf63</rqi>

<ty>4</ty>

<pc>

<m2m:cin>

<cnf>SDT:org.onem2m.home.device/module/temperature </cnf>

<con>

<tempe>

<curT0>5</curT0>

<tarTe>3</tarTe>

</tempe>

</con>

</m2m:cin>

</pc>

</m2m:rqp>

In *content* attribute, a value of DataPoint may be written between tags named the certain DataPoint name.

*contentInfo* attribute is able to be omitted because *content* attribute has tags named the certain Module class name (<temp></temp>).

An example for only DataPoint class mapping is shown below.:

<?xml version="1.0" encoding="UTF-8"?>

<m2m:rqp xmlns:m2m="http://www.onem2m.org/xml/protocols">

<op>1</op>

<to>//example.net/myCSE/-/Cont1</to>

<fr>/myCSE/C2345</fr>

<rqi>0002bf63</rqi>

<ty>4</ty>

<pc>

<m2m:cin>

<cnf>SDT:org.onem2m.home.device/datapoint/temperature/currentTemperature </cnf>

<con>5</con>

</m2m:cin>

</pc>

</m2m:rqp>

In this case, *contentInfo* attribute can NOT be omitted because we cannot understand which Datapoint is written in *content* attribute without *contentInfo* attribute.

If a *contentInfo* attribute is not used, *content* attribute may change as follows:

<con>

<curT0>5</curT0>

</con>

1. History

|  |  |  |
| --- | --- | --- |
| **Publication history** | | |
| V2.0.0 | 2016-08-30 | Release 2 - Publication |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **Draft history** (to be removed on publication) | | |
| V 5.0.0 | 2021-04-06 | TS-0023 adaptation to Release 5 (v4.8.0 🡪 v5.0.0) |
| V5.1.0 | 2021-12-02 | RDM-2021-0067-TS-0023\_Metadata  RDM-2021-0093-SDT-issues-identified\_in\_TDE\_R5 |
| V5.2.0 | 2022-07-15 | Reflecting comments from issues tracker. see link https://git.onem2m.org/issues/issues/-/issues?scope=all&utf8=%E2%9C%93&state=opened&search=rdm |
| V5.3.0 | 2022-09-28 | RDM-2022-0017R03-Proposed\_changes\_to\_TS-0023\_for\_WI\_0109  RDM-2022-0031R01-Adding\_a\_heater\_device\_to\_TS-0023  RDM-2022-0045R01-TS-0023\_PropertyNames\_rel-5  RDM-2022-0051-Adding\_the\_remoteControlEnable\_Module\_to\_the\_deviceRobotCleaner\_Device  RDM-2022-0052R01-Adding\_a\_new\_Device\_model\_deviceJuicer\_to\_TS-0023  RDM-2022-0053R01-Adding\_a\_new\_Device\_model\_deviceShoesWasher\_to\_TS-0023  RDM-2022-0055-Adding\_an\_electric motorcycle\_device\_to\_TS-0023 |