|  |
| --- |
|  |

|  |
| --- |
| CHANGE REQUEST |
| Meeting ID:\* | TST#29 |
| Source:\* | Patricia Martigne, ORANGE, patricia.martigne@orange.com, Maciej Goluch, ORANGE, maciej.goluch@orange.com,Sébastien Bolle, ORANGE, sebastien.bolle@orange.comPawel Srzemecki, ORANGE  |
| Date:\* | 2017-05-18 |
| Reason for Change/s:\* | Add new sections and update essential content of the document |
| CR against: Release\* | Rel-3 |
| CR against: WI\* | [x]  Active <WI-0054> [ ]  MNT maintenance / < Work Item number(optional)>Is this a mirror CR? Yes [ ]  No [ ] mirror CR number: (Note to Rapporteur - use latest agreed revision)[ ]  STE Small Technical Enhancements / < Work Item number (optional)>Only ONE of the above shall be ticked |
| CR against: TS/TR\* | TR-0039 ‘Developer guide: Interworking Proxy using SDT’ |
| Clauses \* | 7, 6.2 |
| Type of change: \* | [ ]  Editorial change[ ]  Bug Fix or Correction[ ]  Change to existing feature or functionality[x]  New feature or functionalityOnly ONE of the above shall be ticked |
| Impacted other TS/TR(s) |  |
| Post Freeze checking:\* | This CR contains only essential changes and corrections? YES [x]  NO [ ] This CR may break backwards compatibility with the last approved version of the TS? YES [ ]  NO [x]  |
| Template Version: January 2017 (Do not modify) |

**oneM2M Notice**

The document to which this cover statement is attached is submitted to oneM2M. Participation in, or attendance at, any activity of oneM2M, constitutes acceptance of and agreement to be bound by terms of the Working Procedures and the Partnership Agreement, including the Intellectual Property Rights (IPR) Principles Governing oneM2M Work found in Annex 1 of the Partnership Agreement.

## Introduction

This document proposes additional content to the developer’s guideline TR-0039 on how to implement SDT and IPE via a simple example. Major changes are made for sections 9 “Implementation” and 6.2 “Introduction to SDT”. It also proposes to add an Annex B for “Short names for resources and attributes ” to extend the short names used in this Developer Guide and make it simpler to understand.

### -----------------------Start of change 1 -------------------------

# Scope

The present document describes how a developer can quickly and easily implement interworking with non-oneM2M devices using the SDT (Smart Device Template) defined in TS-0023 (Home appliances Information Model). As an example, it is proposed a scenario of an application controlling and monitoring a connected lamp, already commercially available on the market. The goal is to describe with an example the methodology for building this interworking, allowing the developers to apply it to any other non-oneM2M devices using the abstraction layer provided by SDT.

To focus on the topics described above security aspect is not considered in the scope of this developer guide, especially Access Control Policy issue is not discussed.

### -----------------------End of change 1---------------------------------------------

-----------------------Start of change 2--------------------------------------------

# 6 Introduction to IPE and SDT

## 6.1 Introduction to IPE

An IPE (Interworking Proxy Entity) is a specialized AE (Application Entity) that allows the oneM2M system to interact with any non-oneM2M system, in a seamless way, through the Mca interface. It has the capability of remapping the specific data model to oneM2M resources (<AE>, <container>, <flexContainer>, etc.) and maintain bidirectional communication with the non-oneM2M system.

## 6.2 Introduction to SDT

### 6.2.1 SDT data model description

The SDT (Smart Home Device Template) is a reference template to model most home appliance functions in a unified way which is a result of consensus amongst various SDOs and industry alliances. Abstraction from the various underlying home-area network technologies and getting an unified way of controlling/commanding the appliances are among the key goals of the SDT.

The SDT approach is to define re-usable basic functions (or services) (labelled "ModuleClass" in Figure 6.2-1 ) which can represent the typical functions found, for example, in many home automation systems, such as "on/off", "dim a lamp", "receive events from binary sensor", "read data from sensor", etc.



Figure 6.2-1: SmartHome Device Template for a generic device

The SDT supports the use of a set of templates for generic devices or appliances (e.g. for a dimmable lamp, a basic washing machine, etc, which would be specific instances of the "Device" object) which form the basis of APIs used by application developers. These templates can also be referenced by manufacturers creating XML documents to describe their specific products. For example, the SDT enables specification of a generic washing machine template, with on/off, set-wash-temperature, pause and a few other commands, which could be referenced by a manufacturer as the schema for a XML description of a basic model washing machine. The SDT allows for vendor-specific additional commands (ModuleClasses) to suit specific product types.

The SDT is available under Apache License 2 at oneM2M’s GitLab: <https://git.onem2m.org/MAS/SDT>

### 6.2.2 SDT Device example: deviceLight

A light is a device that is used to control the state of an illumination device. This Device has one mandatory binarySwitch Module and the following optional Modules: faultDetection, runState, colour, colourSaturation, brightness.

**Table 6.2.2-1: Modules of deviceLight Device model (from [1])**

|  |  |  |  |
| --- | --- | --- | --- |
| Module Instance Name | Module Class Name | Optional | Description |
| faultDetection | faultDetection | true | See clause 5.3.16 |
| binarySwitch | binarySwitch | false | See clause 5.3.5 |
| runState | runState | true | See clause 5.3.28 |
| colour | colour | true | See clause 5.3.10 |
| colourSaturation | colourSaturation | true | See clause 5.3.11 |
| brightness | brightness | true | See clause 5.3.8 |

The faultDetection ModuleClass provides information about whether a fault has occurred in the actual device.

Table 6.2.2-2: DataPoints of faultDetection ModuleClass (from [1])

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | Readable | Writable | Optional | Documentation |
| status | xs:boolean | true | false | false | Status of fault detection. |
| code | xs:integer | true | false | true | Code of the fault. |
| description | xs:string | true | false | true | Message of the fault. |

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

Table 6.2.2-3: Actions of binarySwitch ModuleClass (from [1])

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

Table 6.2.2-4: DataPoints of binarySwitch ModuleClass (from [1])

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | Readable | Writable | Optional | Documentation |
| powerState | xs:boolean | true | true | false | The current status of the binarySwitch. "True" indicates turned-on, and "False" indicates turned-off. |

The runState ModuleClass provides capabilities to control and monitor machine state of appliances.

Table 6.2.2-5: DataPoints of runState ModuleClass (from [1])

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | Readable | Writable | Optional | Documentation |
| currentMachineState | hd:machineState | true | true | false  | Currently active machine state. |
| machineStates | list of hd:machineState | true  | false | false | List of possible machine states the device supports (see clause 5.5.15) |
| progressPercentage | float | true | false | true | Indication of current progress in percentage |

The colour ModuleClass provides the capabilities to set the value of Red, Green, Blue for the colour device.

Table 6.2.2-6: DataPoints of colour ModuleClass (from [1])

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | Readable | Writable | Optional | Documentation |
| red | xs:integer | true | true | false | The R value of RGB; the range is [0,255] |
| green | xs:integer | true | true | false | The G value of RGB; the range is [0,255] |
| blue | xs:integer | true | true | false | The B value of RGB; the range is [0,255] |

The coourSaturation ModuleClass describes a colour saturation value. The value is an integer. A colourSaturation has a range of [0,100]. A colourSaturation value of 0 means producing black and white images. A colourSaturation value of 50 means producing device specific normal colour images. A colourSaturation value of 100 means producing device very colourfull images.

Table 6.2.2-7: DataPoints of colourSaturation ModuleClass (from [1])

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | Readable | Writable | Optional | Documentation |
| colourSaturation | xs:integer | true | true | false | The status of colour saturation level. |

The brightness ModuleClass describes the brightness of a light e.g. 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 6.2.2-8: DataPoints of brightness ModuleClass (from [1])

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Name | Type | Readable | Writable | Optional | Documentation |
| brightness | xs:integer | true | true | false | The status of brightness level in percentage. |

### -----------------------End of change 2---------------------------------------------

-----------------------Start of change 3--------------------------------------------

# 9 Implementation

## 9.1 Introduction

Clause 9 provides implementation examples from two different perspectives: developer of a device adapter (AE: Inter-working Proxy) in clause 9.3 and developer of an utility application in clause 9.4.

### **oneM2M SDT Hue**

**Figure 9.1-1 : implementation of TS-0023 module classes for the Hue bulb use case**

* 9.2 AssumptionsMN-CSE is registered with IN-CSE
* Philips Hue Bridge and MN-CSE are on the same LAN
* Philips Hue Bulbs are paired with the Philips Hue Bridge
* Philips Hue Bridge is not directly represented in oneM2M resource tree, but through its associated IPE
* Access Control Policy issue isn’t discussed here because it's a security aspect which isn’t considered in the scope of this developer guide

### 9.2.1 Addresing for Entities

Each oneM2M entity including AE and CSE are addressable with correct host address that can be IP addresses or FQDN addresses resolved to IP addresses by DNS network services according to addressing rules specified in oneM2M standards.

The IN-CSE and MN-CSE entities presented in this use case are addressable with the following identifiers.

* IN-CSE:
	+ CSE-ID: /**in-cse**
	+ resourceName of IN-CSE’s CSEBase resource: **server**
	+ IN-CSE FQDN**: incse.provider.com**
	+ IN-CSE HTTP port: **8080**
* MN-CSE:
	+ CSE-ID: /**mn-cse**
	+ resourceName of MN-CSE’s CSEBase resource: **home\_gateway**
	+ MN-CSE FQDN: **mncse.provider.com**
	+ MN-CSE HTTP port: **8080**

## 9.3 Developer of the device adapter (IPE-AE)

This clause describes implementation process from the IPE-AE developer point of view.



Figure 9.3-1 Scope of 9.3 Developer of the device adapter section

### 9.3.1 IPE-AE registration with MN-CSE

The IPE-AE with MN-CSE registration is shown in the following procedure.

POST /home\_gateway HTTP/1.1
Host: mncse.provider.com:8080
X-M2M-Origin: Soriginator

Content-Type: application/xml;ty=2

X-M2M-RI: home\_gateway-16346

<?xml version="1.0" encoding="UTF-8"?>

<m2m:ae xmlns:m2m=<http://www.onem2m.org/xml/protocols> rn="ipe\_ae">

 <api>ipe.lightControl</api>

 <rr>true</rr>

</m2m:ae>

HTTP Response:

201 Created

X-M2M-RSC: 2001

X-M2M-RI: home\_gateway-16346

Content-Location: /mn-cse/ae-CAE340304178

In response there's Content-Location header which indicates registrated AE address which is necessary to create device resource tree.

AE registration process is comprehensively described in 10.2.2.1 TS-0001 clause.

### 9.3.2 SDT Device resource tree creation in MN-CSE (Resources registration)

This cluase describes creation of device resource tree which consists of Device, ModuleClasses and Actions. First of all you have to create Device resource Device is mapped to flexContainer resource, so you have to create flexContainer resource.

The following request creates flexContainer resource for deviceLight. In response there's Content-Location header which indicates the new Device address which is used later in ModuleClass creation.

POST /~/mn-cse/ae-CAE340304178 HTTP/1.1
Host: mncse.provider.com:8080
X-M2M-Origin: Soriginator

Content-Type: application/xml;ty=28

X-M2M-RI: home\_gateway-12

<?xml version="1.0" encoding="UTF-8"?>

<m2m:fcnt xmlns:m2m=<http://www.onem2m.org/xml/protocols> rn="deviceLight">

 <cnd>org.onem2m.home.device.deviceLight</cnd>

</m2m:fcnt>

HTTP Response:

201 Created

X-M2M-RSC: 2001

X-M2M-RI: home\_gateway-12

Content-Location: /mn-cse/DL3404345178

Flex Container creation procedure is comprehensively described in 10.2.4.16 TS-0001.

To create ModuleClass on Device you have to send the creation request to Device address which was taken from Device creation request response. In SDT ModuleClasses are mapped to flexContainer, Datapoints are mapped to customAttribute. To create binaryswitch ModuleClass you have to send following request.

POST /~/mn-cse/DL3404345178 HTTP/1.1
Host: mncse.provider.com:8080

X-M2M-Origin: Soriginator

Content-Type: application/xml;ty=28

X-M2M-RI: home\_gateway-43

<?xml version="1.0" encoding="UTF-8"?>

<m2m:fcnt xmlns:m2m=<http://www.onem2m.org/xml/protocols> rn="binarySwitch">

 <cnd>org.onem2m.home.moduleclass.binaryswitch</cnd>

 <powSe type="xs:boolean">false</powSe>

</m2m:fcnt>

HTTP Response:

201 Created

X-M2M-RSC: 2001

X-M2M-RI: home\_gateway-43

Content-Location: \_/mn-cse/MC43546456

In response there's Content-Location header which indicates created ModuleClass address which is used later to create Toggle Action and to modificate state of particular DataPoint.

In SDT Actions are also mapped to FlexContainer. To add new Action to ModuleClass you have to send the following request to the ModuleClass address.

POST /~/mn-cse/MC43546456 HTTP/1.1
Host: mncse.provider.com:8080

X-M2M-Origin: originator

Content-Type: application/xml;ty=28

X-M2M-RI: home\_gateway-44

<?xml version="1.0" encoding="UTF-8"?>

<m2m:fcnt xmlns:m2m=<http://www.onem2m.org/xml/protocols> rn="toggle">

<cnd>org.onem2m.home.moduleclass.binaryswitch.toggle</cnd>

</m2m:fcnt>

HTTP Response:

201 Created

X-M2M-RSC: 2001

X-M2M-RI: home\_gateway-44

Content-Location: /mn-cse/AC-54783722

In response there's Content-Location header which indicates new Action address which is used later to trigger Action (described in 9.4.3 clause).

### -----------------------End of change 3---------------------------------------------

-----------------------Start of change 4--------------------------------------------

## 9.4 Developer of the utility application

This clause describes registration of AE for utility application, discovery process and controlling and monitoring devices.

###

Figure 9.4-1 Scope of 9.4 Developer of the utility application section

### 9.4.1 Application Entity registration in IN/MN-CSE

The ADN-AE with IN-CSE registration is shown in the following procedure.

POST /server?rcn=0 HTTP/1.1
Host: incse.provider.com:8080
X-M2M-Origin: Soriginator

Content-Type: application/xml;ty=2

X-M2M-RI: server-1234

<?xml version="1.0" encoding="UTF-8"?>

<m2m:ae xmlns:m2m=<http://www.onem2m.org/xml/protocols> rn="smartphone\_ae">

<api>incse.lightControlApp</api>

<rr>true</rr>

</m2m:ae>

HTTP Response:

201 Created

X-M2M-RSC: 2001

X-M2M-RI: server-1234

Content-Location: /in-cse/ae-CAE345

In response Content-Location header there's an address of registered Application Entity.

AE registration process is comprehensively described in 10.2.2.1 TS-0001 clause.

### 9.4.2 Discovery proccess

This clause describes discovery proccess which uses filter criteria to limit the scope of returned information.

Folowing response searches for resources with cnd(Container Definition) set to org.onem2m.home.device.deviceLight so it gives in response list of urils of light devices. Uril is neccessary to control particular device. Fu (filter usage) parameter value set to '1' indicates this is normal resource discovery request so only the addresses of the child resources are returned.

GET /server?fu=1&cnd=org.onem2m.home.device.deviceLight HTTP/1.1

Host: incse.provider.com:8080
X-M2M-Origin: Soriginator

Content-Type: application/xml

X-M2M-RI: server-1234

HTTP Response:

200 OK

X-M2M-RSC: 2000

X-M2M-RI: server-1234

Content-Type: application/xml

<m2m:uril xmlns:m2m="http://www.onem2m.org/xml/protocols">

mn-cse/DL3404345178

</m2m:uril>

Discovery process is comrehensively described in TS-0001 6.2.5 and 10.2.6 clause.

### 9.4.3 Control & monitor devices

This clause describes how to control and monitor SDT devices registered in oneM2M network.

First there is a need to get an url of ModuleClass resource. ModuleClass is a child of deviceLight, so send request to device address which returns a list of it's children in response. In this particular case there're two: binarySwitch and toggle. Each of them shall be controlled by requests sent on given addresses.

Following request gives a list of deviceLight children (ModuleClassess or Actions) in response with their urils.

GET /~/mn-cse/DL3404345178?rcn=6

HTTP/1.1
Host: incse.provider.com:8080
X-M2M-Origin: Soriginator

Content-Type: application/xml;

X-M2M-RI: server-12345

HTTP Response:

200 OK

X-M2M-RSC: 2000

X-M2M-RI: server-12345

Content-Location: /in-cse/ae-CAE345

<?xml version="1.0" encoding="UTF-8"?>

<m2m:fcnt xmlns:m2m="http://www.onem2m.org/xml/protocols">

 <ch rn="binarySwitch " ty="28">/mn-cse/MC43546456 </ch>

<ch rn="toggle " ty="28">/mn-cse/AC-54783722 </ch>

</m2m:fcnt>

Here is data from previous request answer which is needed:

<ch rn="binarySwitch " ty="28">/mn-cse/MC43546456 </ch>

<ch rn="toggle " ty="28">/mn-cse/AC-54783722 </ch>To change powerState value you have to send a request to binarySwitch resource (/mn-cse/MC43546456 )with xml (or json) representation of new powerState value. In response there's HTTP 201 response code which means created status. PUT request is used because whole resoruce is updated.

Following requests changes power state data point to 'true'.

PUT /~/mn-cse/MC43546456 HTTP/1.1
Host: incse.provider.com:8080
X-M2M-Origin: Soriginator

Content-Type: application/xml;

X-M2M-RI: server-12346

<?xml version="1.0" encoding="UTF-8"?>

<m2m:fcnt xmlns:m2m=<http://www.onem2m.org/xml/protocols> rn="smartphone\_ae">

<powSe type="xs:boolean">true</powSe>

</m2m:fcnt>

HTTP Response:

204 Updated

X-M2M-RSC: 2004

X-M2M-RI: server-12346

To trigger Toggle action you have to send null content parameter to toggle resource address. (see TS-0023, 6.2.4 Clause, Rule 3-4).

Following requests sends null content and triggers Toggle action.

PUT /~/mn-cse/AC-54783722 HTTP/1.1
Host: incse.provider.com:8080
X-M2M-Origin: Soriginator

Content-Type: application/xml;

X-M2M-RI: server-12346

<?xml version="1.0" encoding="UTF-8"?>

<m2m:fcnt xmlns:m2m=<http://www.onem2m.org/xml/protocols> rn="smartphone\_ae">

</m2m:fcnt>

HTTP Response:

204 Updated

X-M2M-RSC: 2004

X-M2M-RI: server-12346

### -----------------------End of change 4---------------------------------------------

### -----------------------Start of change 5-------------------------------------------

Annex A:

Further readings proposition

oneM2M TR-0017 Home Domain Abstract Information Model

oneM2M TR-0025 Application Developer Guide

…

Annex B:

Short names for resources and attributes

### -----------------------End of change 5---------------------------------------------