RFC 235 - Abstraction Layer with oneM2M Smart Device Template

Draft

96 Pages

Abstract

This document describes the need of using the oneM2M device abstraction layer in an OSGi platform, a comparison with the current OSGi Device Abstraction Layer, the major requirements for the integration of oneM2M standard concepts and a technical solution for an implementation on OSGi technology.

0 Document Information

0.1 License

DISTRIBUTION AND FEEDBACK LICENSE, Version 2.0

The OSGi Alliance hereby grants you a limited copyright license to copy and display this document (the “Distribution”) in any medium without fee or royalty. This Distribution license is exclusively for the purpose of reviewing and providing feedback to the OSGi Alliance. You agree not to modify the Distribution in any way and further agree to not participate in any way in the making of derivative works thereof, other than as a necessary result of reviewing and providing feedback to the Distribution. You also agree to cause this notice, along with the accompanying consent, to be included on all copies (or portions thereof) of the Distribution. The OSGi Alliance also grants you a perpetual, non-exclusive, worldwide, fully paid-up, royalty free, limited license (without the right to sublicense) under any applicable copyrights, to create and/or distribute an implementation of the Distribution that: (i) fully implements the Distribution including all its required interfaces and functionality; (ii) does not modify, subset, superset or otherwise extend the OSGi Name Space, or include any public or protected packages, classes, Java interfaces, fields or methods within the OSGi Name Space other than those required and authorized by the Distribution. An implementation that does not satisfy limitations (i)-(ii) is not considered an implementation of the Distribution, does not receive the benefits of this license, and must not be described as an implementation of the Distribution. "OSGi Name Space" shall mean the public class or interface declarations whose names begin with "org.osgi" or any recognized successors or replacements thereof. The OSGi Alliance expressly reserves all rights not granted pursuant to these limited copyright licenses including termination of the license at will at any time.

EXCEPT FOR THE LIMITED COPYRIGHT LICENSES GRANTED ABOVE, THE OSGi ALLIANCE DOES NOT GRANT, EITHER EXPRESSLY OR IMPLIEDLY, A LICENSE TO ANY INTELLECTUAL PROPERTY IT, OR ANY THIRD PARTIES, OWN OR CONTROL. Title to the copyright in the Distribution will at all times remain with the OSGi Alliance. The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted therein are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

THE DISTRIBUTION IS PROVIDED "AS IS," AND THE OSGi ALLIANCE (INCLUDING ANY THIRD PARTIES THAT HAVE CONTRIBUTED TO THE DISTRIBUTION) MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE DISTRIBUTION ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.
NEITHER THE OSGi ALLIANCE NOR ANY THIRD PARTY WILL BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR RELATING TO ANY USE OR DISTRIBUTION OF THE DISTRIBUTION.

Implementation of certain elements of this Distribution may be subject to third party intellectual property rights, including without limitation, patent rights (such a third party may or may not be a member of the OSGi Alliance). The OSGi Alliance is not responsible and shall not be held responsible in any manner for identifying or failing to identify any or all such third party intellectual property rights.

The Distribution is a draft. As a result, the final product may change substantially by the time of final publication, and you are cautioned against relying on the content of this Distribution. You are encouraged to update any implementation of the Distribution if and when such Distribution becomes a final specification.

The OSGi Alliance is willing to receive input, suggestions and other feedback (“Feedback”) on the Distribution. By providing such Feedback to the OSGi Alliance, you grant to the OSGi Alliance and all its Members a non-exclusive, non-transferable,

worldwide, perpetual, irrevocable, royalty-free copyright license to copy, publish, license, modify, sublicense or otherwise distribute and exploit your Feedback for any purpose. Likewise, if incorporation of your Feedback would cause an implementation of the Distribution, including as it may be modified, amended, or published at any point in the future ("Future Specification"), to necessarily infringe a patent or patent application that you own or control, you hereby commit to grant to all implementers of such Distribution or Future Specification an irrevocable, worldwide, sublicenseable, royalty free license under such patent or patent application to make, have made, use, sell, offer for sale, import and export products or services that implement such Distribution or Future Specification. You warrant that (a) to the best of your knowledge you have the right to provide this Feedback, and if you are providing Feedback on behalf of a company, you have the rights to provide Feedback on behalf of your company; (b) the Feedback is not confidential to you and does not violate the copyright or trade secret interests of another; and (c) to the best of your knowledge, use of the Feedback would not cause an implementation of the Distribution or a Future Specification to necessarily infringe any third-party patent or patent application known to you. You also acknowledge that the OSGi Alliance is not required to incorporate your Feedback into any version of the Distribution or a Future Specification.

I HEREBY ACKNOWLEDGE AND AGREE TO THE TERMS AND CONDITIONS DELINEATED ABOVE.

0.2 Trademarks

OSGi™ is a trademark, registered trademark, or service mark of the OSGi Alliance in the US and other countries. Java is a trademark, registered trademark, or service mark of Oracle Corporation in the US and other countries. All other trademarks, registered trademarks, or service marks used in this document are the property of their respective owners and are hereby recognized.

0.3 Feedback

This document can be downloaded from the OSGi Alliance design repository at https://github.com/osgi/design
The public can provide feedback about this document by opening a bug at https://www.osgi.org/bugzilla/.

0.4 Table of Contents

0.5 Terminology and Document Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY" and "OPTIONAL" in this document are to be interpreted as described in 10.1.

`Source code is shown in this typeface.`

0.6 Revision History

The last named individual in this history is currently responsible for this document.

Revision	Date	Comments
Initial	Jun 08 2017	Initial draft version. Maciej Goluch, Orange Polska S.A.
0.01	Jun 12 2017	Change structure of the document. Provide Introduction chapter, improve Device, ModuleClass, Datapoint chapters. Andre Bottaro, Orange
0.02	Jun 14 2017	Developed initial draft version of interfaces for ModuleClass, Device, DataPoint, Domain, Doc, Property. Pawel Strzemecki, Orange Polska S.A.
0.03	Jun 28 2017	Review and improve the code and developed all others interfaces and classes. Provide Entities, Essentials, Operation Summary chapters and improve Device, ModuleClass, DataPoint chapters. Maciej Goluch, Orange Polska S.A.

1 Introduction

OSGi technology is more and more popular in residential and Internet of Things solutions. Many devices use various protocols on the market which lead to the development of complex smart systems. In order to provide a convenient programming model, to make the smart solutions integration with these devices easier, the OSGi

Alliance has specified and published a specification named Device Abstraction Layer within OSGi Residential Specification Release 6.

However, the OSGi Device Abstraction Layer adds a new abstraction model that is not backed by any software community or standard organization. The main concepts of the OSGi current Device Abstraction layer are close to the ones of the Smart Device Template made by the Home Gateway Initiative and now adopted by oneM2M organization [1]. Adopting the Smart Device Template (SDT) is an opportunity to reach the audience of a strong organization, whose specifications are being implemented by open source communities.

This RFC describes the need to back OSGi Device Abstraction Layer with oneM2M SDT standard, compares both models, defines the requirements and defines how to integrate this solution with OSGi Alliance.

2 Application Domain

Currently, IoT devices use various communication protocols. To provide a good support for third party developers, many organizations and companies create their own abstraction model of devices. So many abstraction layers exist in IoT application domain.

The application domain is well described in RFP 147 Error: Reference source not found. The picture from the RFP 147 on Fig 1 shows a reference architecture which is valid in the majority of cases under consideration.

The existing OSGi specifications which address related topics are:

Device Access Specification – focuses on the dynamic discovery of the proper driver when a new device is attached/connected to the residential gateway. The specification is limited to the driver installation needs.

UPnP™ Device Service Specification – defines an OSGi API to work with UPnP devices accessible on the home network

EnOcean Device Service Specification – defines an OSGi API to work with EnOcean devices accessible on the home network.

Device Abstraction Layer – unifies the work with devices provided by different protocols.

Device Abstraction Layer Functions Specification – defines concrete device function interfaces

2.1 Terminology + Abbreviations

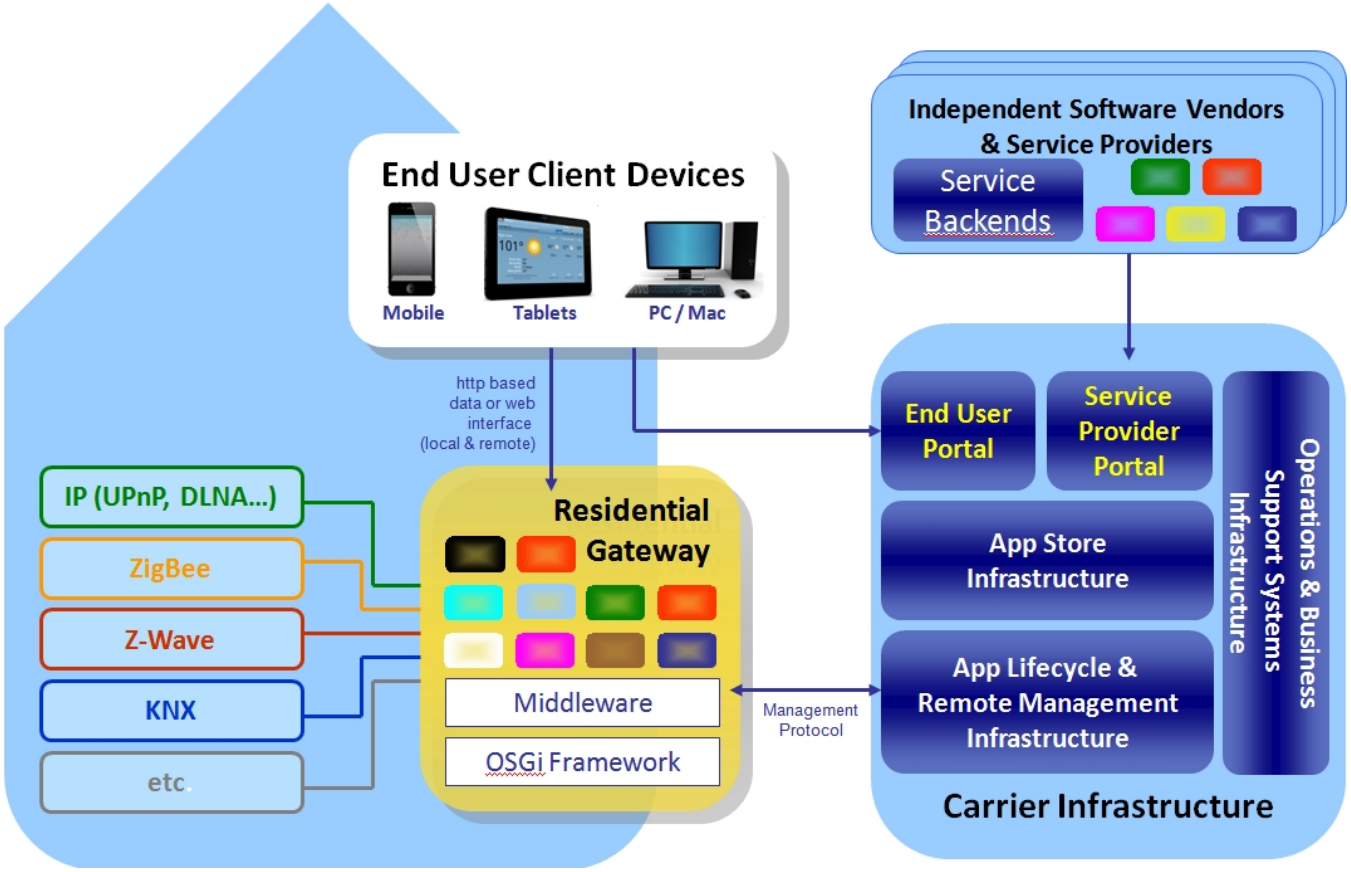

Fig 1: Abstraction layer in the Smart Home environment

Item	Description
Device Abstraction Layer	Unifies the work with devices provided by different protocols.
Smart Device Template	The SDT (Smart Device Template) is an open specification started by the Home Gateway Initiative (HGI) to find consensus amongst various SDOs and industry alliances to derive a common approach for device modeling.. Its further development is now governed by oneM2M and hosted by ETSI.

3 Problem Description

Today there are many various communication protocols which IoT devices use to exchange information. In a common IoT system architecture, a piece of middleware software provides at least the connection between the devices using various communication protocols. To provide more convenience in application development process companies raise the level of programming abstraction through implement abstraction models of devices and theirs functions. The advantage of this approach is the application developers do not need to pay attention on a communication layer, they just focus on a businesses logic.

Many organizations and companies have created their own abstraction model of devices. Disunited solutions without common information model brings to a lack of interoperability between these systems. Today, third party application developers have to implement several abstraction layers to allow their application to be platform independent. They especially have to implement oneM2M Smart Device Template and information model which are now emerging as reference technologies in the IoT domain.

In order to avoid this situation and provide more convenient and standard programming model, the OSGi Alliance should adopt a standard device model, which is agreed by almost all IoT and M2M players. It could be achieved by using the standard abstraction model defined and manage by oneM2M, Smart Device Template (SDT). SDT is an open source project (under Apache License, version 2.0) and publicly accessible Error: Reference source not found An additional document (TR-0017 Error: Reference source not found) helps in the use of SDT.

One aim of the OSGi Device Abstraction layer (DAL) was initially to converge with the information model of the Home Gateway initiative (see OSGi RFP 147) that has been later adopted by oneM2M. The Home Gateway initiative has however specified the Smart Device Template after the OSGi Alliance has finalized its own layer. DAL and SDT specifications are now distinct. They fortunately share many high-level concepts while some of them and many details differ. The table below shows a comparison between the two models.

SDT	DAL
Domain • A unique name which acts like a namespace (e.g., "org.oneM2M.home.modules"). It is set by the organization creating the model. • Allowing reference to a package of definitions for the contained ModuleClasses and Device models.	**No equivalent concept.** The Java package naming is an opportunity to distinguish namespaces, which is not seized by the current OSGi DAL specification.
ModuleClass • Specifies a single service (e.g., audioVolume, binarySwitch) with one or more Actions, Properties, DataPoints and Events. • Every service that is described as a ModuleClass can be re-used in many Devices.	**Function** • Atomic functional entity like switch or sensor. • The function belongs to a device. • The function provides a set of properties and operations
Module • Instantiation of a ModuleClass for a specific Device or SubDevice.	**Function subclass** • Subclass of a Function to implement specific functionalities.
Device • Physical, addressable, identifiable appliance, sensor and actuator • Contains one or more Modules, Properties and SubDevices	**Device** • Represents the device in the OSGi service registry • It is described with a set of service properties and functions.
SubDevice • SubDevice is a device which may be embedded in a Device and/or is addressed via another Device.	There is no equivalent concept in OSGi DAL. It might be mappable with the Device and keep relationship through service methods and service properties, like it is done for UPnPDevice services and their device

	children., similar to the *SERVICE_REFERENCE_UIDS* property in the Function interface.
DataPoint • DataPoint shall be used to represent stateless operations • Functional information • E.g. targetTemperature in a thermostat, targetVolume in a television.	**FunctionData or Function Properties** • Data structure which carries Function property value with extra metadata. • To be noted: an ambiguity between Functiondata and function property in OSGi DAL.
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 a user name and a password at the same time • As a result, one or more DataPoints can be updated.	**Function Operations** • Function operations are the main callable units. • They can be used by the applications to control the device • Synchronous or asynchronous operations can trigger different actions • As a result, a Function property can be updated
Arg • Represents the parameter information which a device needs to carry out in a required "Action". • An Action may have none, one, or more Args.	**Java methods arguments** • Function Operations have zero or more arguments and zero or one return value.
Property • Device, SubDevice or ModuleClass could have Properties • Non-functional information • E.g. version, id and etc.	**Function service properties and Device service properties** • services can be filtered and accessed with their properties • Allows to define non-functional prop.: service name, version etc. and functional properties for Devices: status. • Can be used to represent different relationship between devices
DataType Metadata • DataTypes, used in DataPoints, Args etc support *unitOfMeasure* and *constraints* to express the specification of units as well as upper/lower limits, etc.	**PropertyMetadata** • contain metadata about the Function properties **OperationMetadata** • contain metadata about the Function operations
Event • elements are needed for automation protocols which "push" information, instead of relying on polling by the software application. • can be signalled ("published") by devices asynchronously	**FunctionEvent** • Asynchronous event. • It is posted through EventAdmin service and notifies for Function property change.
DataType • can be simple integers or string text, or rather complex • SimpleType • Boolean, Byte, Integer, Float, String, Enum, Date, Time, Datetime, Blob, Uri • Struct	**Types in DAL** • Java primitive type or corresponding reference type. • java.lang.String. • Numerical type i.e. the type which extends java.lang.Number. • Beans, but the beans properties must use

• Array	those rules. Java Beans are defined in JavaBeans specification. • java.util.Maps. The keys can be java.lang.String. The values of a single type follow these rules. • Arrays of defined types.
Potential possibility to represent the OSGi lifecycle using CRUD operations in oneM2M. The SDT is a meta-format to express data models, not the lifecycle of devices and functionalities.	**Lifecycle** • There is a possibility to install,/uninstall, start/stop bundles. • The bundles could dynamically register/unregister services. • Function services are registered before the Device service
Device models • Definition of devices, e.g. deviceLight, deviceOven	There is no particular devices specification in DAL, only a specification of basic functions exist.
Doc • contain the human-readable information • Each elements could have a Doc description	Documentation of the elements is provided in the Chapter 142 of the main specification document

The Smart Device Template is depicted with a UML diagram in oneM2M specification documents, see Error: Reference source not found.

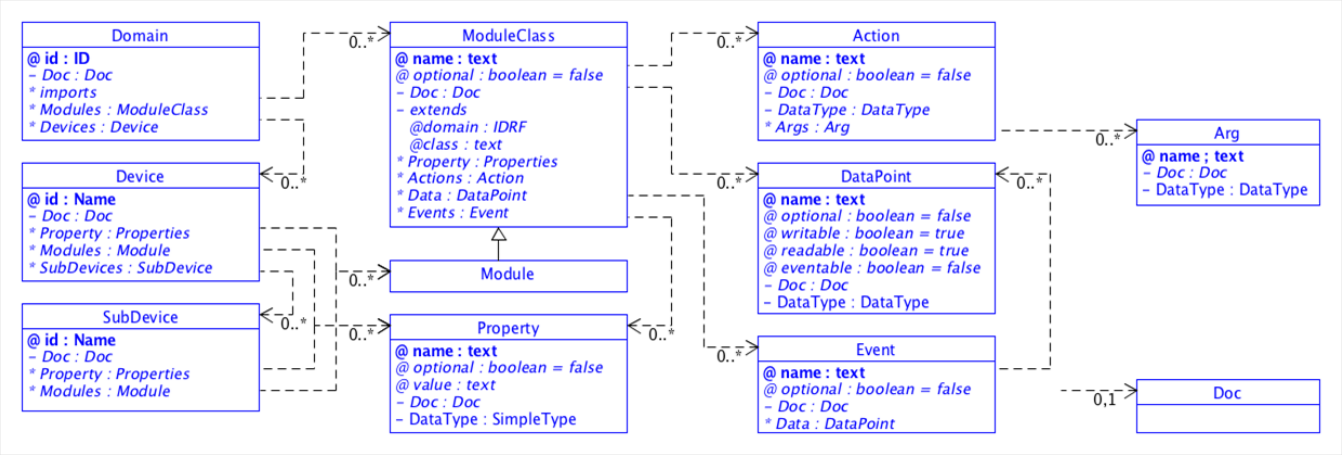

Fig 2: Smart Device Template diagram

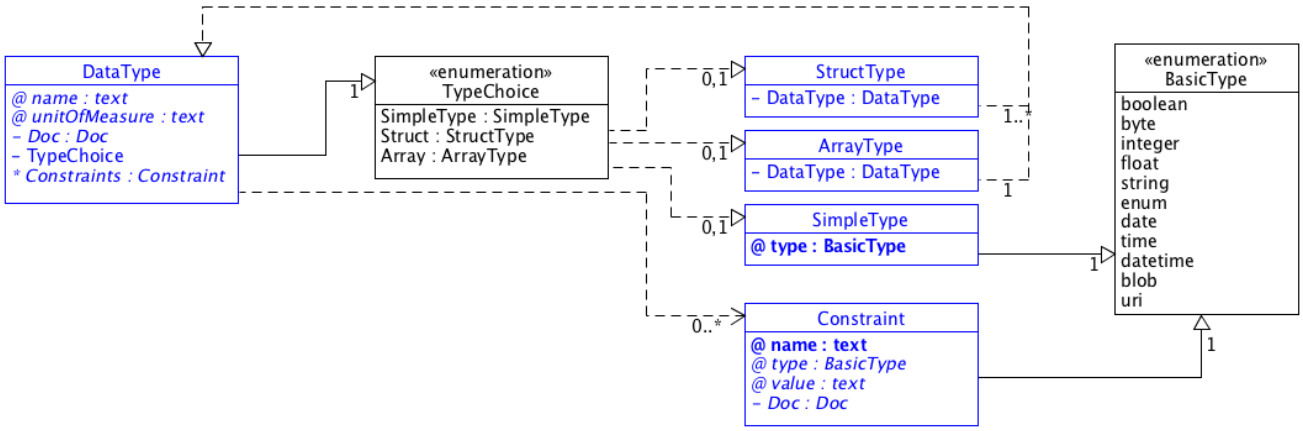

Fig 3: Smart Device Template data types

To sum up the best idea is to provide a solution which is common, agreed by almost all of IoT players and provide many devices and reusable device functions models specifications. It could be all done by implementing the Smart Device Template and taking all advantages of it, including a set of Device and ModuleClass models specifications. The picture in Error: Reference source not foundshows a general possible architecture of implementation which is based on the architecture proposed in RFP-147.

4 Requirements

Requirement 1. The solution MUST support SDT concepts to represent abstract device and function classes.
Requirement 2. The solution MUST enable the discovery and control of devices which is applicable for any relevant device protocol.
Requirement 3. The solution MUST define an API for the control of devices, which is independent from the device protocols.
Requirement 4. The solution MUST include device access control based on user and application permissions compliant with the OSGi security model.
Requirement 5. The solution MUST take advantage of the security features available in the device protocols.
Requirement 6. The solution MUST include a device protocol independent notification mechanism realized according to the OSGi event mechanisms.
Requirement 7. The solution MUST map all oneM2M SDT concepts.
Requirement 8. The solution MUST enable the implementation of oneM2M home information model (TS-0023).
Requirement 9. The solution MUST register device or/and device related instances to the OSGi service registry.
Requirement 10. The solution MUST represent oneM2M domain concept.
Requirement 11. The solution MUST be independent from OSGi Device Abstraction Layer.

5 Technical Solution

5.1 Introduction

The heterogeneity is one main challenge the applications developers are facing in the Internet of Things. To ease their job, they look for a so-called device abstraction layer. This is a software layer that implements a generic set of interfaces to represent any device whatever the technology is used by the device. Thanks to this layer, applications discover, describe and control heterogeneous devices with a unique set of interfaces whereas the set of device protocols is plural and extensible.

OneM2M is a large international standard organization that has specified a set of device represenations with a unique device data model named the Smart Device Template (SDT, [1]). This RFC is mapping oneM2M SDT concepts into Java programming language and OSGi technology. The implementation of this RFC is an extensible device abstraction layer using oneM2M standard data model.

With the use of this abstraction layer, an OSGi application will discover, describe and control devices by requesting OSGi devices and modules in a format that is mirroring oneM2M SDT and set of device representations, e.g, the representation of a light, a coffee machine, an oven. The core data model and the set of device representations will evolve with the work of oneM2M working groups.

5.1.1 Essentials

- *Scope* – This specification is limited to implementation of abstract data model and can be considered as template. Aspects concerning the representation of particular SDT ModuleClasses or Devices are not addressed (it could be covered by a separate RFC).
- *OSGi Services for Device and ModuleClasses* – Device Service with associated Module Services MUST be registered on the same BundleContext object. Device Service shall be registered before Module Services registration.

5.1.2 Entities

Domain - Domain is a unique name which acts like a namespace (e.g., "org.osgi.sdt.home.modules"). It allows to reference to a package of definitions for the contained ModuleClasses and Device models.

Device - Device model is a physical, addressable, identifiable appliance, sensor and actuator with one or more Modules, Properties and SubDevices.

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

Datapoint – DataPoint is designed for stroing functional information (E.g. functional information: targetTemperature, targetVolume).

Property – Property is designed for storing non-functional information (E.g. non-functional information: version, id).

DataType – SDT provides a set of DataTypes. Data can be represented as simple integer, string or more complex structures like SDT Array or SDT Struct

5.1.3 Operation Summary

To simplify the generic features discovery, both the Device interface and the particular SDT Device interface MUST be used for the Device Service registration. Similarly, both the ModuleClass interface and the particular ModuleClass interface MUST be used for the Module Service registration. Both Device Service and Module Service MUST be registred on the same BundleContext object.

5.2 Domain

Domain is a Java interface representing SDT Domain. It has a set of methods allowing to manage Java Collections dedicated for SDT ModuleClasses and SDT Devices.

5.3 Device

Device is a service interface representing a device with oneM2M SDT. A Device includes ModuleClasses objects and Properties linked to this interface with getters. Every Device MUST be registered with associated Properties in the service registry.

SDT Device is represented as OSGi Service. Device Service shall be registered before Module Services registration. Device service MUST be registered with Device interface name and the name of particular SDT Device interface (e.g. for SDT deviceLight there could be Light Java interface provided). Device Service and associated with it Module Services MUST be registered on the same BundleContext object.

The Device service has a rich set of properties (some of them are optional).

Mandatory properties:

- `Device.LABEL – (sdt.device.label/String).` Unique device label assigned by the manufacturer. The uniqueness may be global or only valid within a certain domain (e.g. vendor-wise or for a certain *`Device.Type`*). It's typically used as e.g. device serial number.
- `Device.TYPE` – (`sdt.device.type/String`). The type (e.g. cell phone, photo frame, smart meter) or product class (e.g. X-series) of the device.
- `Device.MODEL` – (`sdt.device.model/String`). The name/identifier of the device mode assigned by the manufacturer.
- `Device.MANUFACTURER – (sdt.device.manufacturer/String).` The name/identifier of the device manufacturer.
- `Constants.SERVICE_PID – (service.pid/String).` Service PID value follows the following form:

  "<containerDefinition value>.<`Device.MANUFACTURER` value>.<`Device.MODEL` value>.<ID>"

  Where <containerDefinition value> is a name of specializations for device models defined in [1] (E.g. containerDefinition for deviceLight is "org.onem2m.home.device.deviceLight") and <ID> is a unique number defined by the developer. The uniqueness may be global or only valid within a "<containerDefinition value>.<`Device.MANUFACTURER` value>.<`Device.MODEL` value>" namespace.
- `Device.DEVICE_CATEGORY – (DEVICE_CATEGORY/String).` The Device service MUST also be registered with `org.osgi.service.device.Constants.DEVICE_CATEGORY` property (see OSGi Compendium: 103 Device Access Specification) that describes a table (String[]) of categories to which the device belongs. One value MUST be "SDT".

Optional properties:

- `Device.NAME` – (`sdt.device.name/String`). The name of the device.
- `Device.MANUFACTURER_DETAILS_LINK – (sdt.device.manufacturer.link/String).` URL to manufacturer's website.

- `Device.MANUFACTURING_DATE` – (`sdt.device.manufacturer.date/String`). Manufacturing date of device.
- `Device.SUB_MODEL` – (`sdt.device.submodel/String`). Device sub-model name.
- `Device.FW_VERSION` – (`sdt.device.version.fw/String`). The firmware version of the device.
- `Device.SW_VERSION` – (`sdt.device.version.sw/String`). The software version of the device.
- `Device.HW_VERSION` – (`sdt.device.version.hw/String`). The hardware version of the device.
- `Device.OS_VERSION` – (`sdt.device.version.os/String`). Version of the operating system (defined by manufacturer).
- `Device.COUNTRY` – (`sdt.device.country/String`). Country code of the device. It could be manufacturing country, deployment country or procurement country.
- `Device.LOCATION` – (`sdt.device.location/String`). Location where the device is installed. It may be configured via the user interface provided by the 'presentationURL' property or any other means.
- `Device.SYSTEM_TIME` – (`sdt.device.system.time/String`). Reference time for the device.
- `Device.SUPPORT_URL` – (`sdt.device.support.url/String`). URL that points to product support information of the device.
- `Device.PRESENTATION_URL` – (`sdt.device.presentation.url/String`). 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".

5.4 ModuleClass

ModuleClass is a Java interface representing SDT ModuleClass. A ModuleClass includes Actions, Properties, DataPoints and Events linked to this interface with getters. A ModuleClass includes also getter and setter to manage optionality of particular ModuleClass in particular Device.

ModuleClass interface has also a setter and a getter for name of an instance of object implementing ModuleClass interface. If SDT Device has more than one instance of the same ModuleClass the instance name MUST be used to distinguish the instances. The instance name is corresponding to definition in [1] (e.g. waterStatus) and referenced by a getter in the parent Device class (e.g. DeviceCoffeeMachine).

Every ModuleClass MUST be registered in OSGi Service Registry as a Module Service. Their properties (if any) shall be registered as OSGi service properties. Module Service shall be registered after Device service registration. Device Service and associated with it Module Services MUST be registered on the same BundleContext object. Module Service MUST be registered with ModuleClass interface name and the name of particular SDT ModuleClass interface (e.g. for SDT binarySwitch ModuleClass there could be BinarySwitch Java interface provided).

5.5 DataPoint

DataPoint is a Java class which is inherited thorugh abstract class ValuedDataPoint. For each type of SDT Datapoint there is a seperate Java class which extends ValuedDataPoint (E.g. IntegerDataPoint inherits from ValuedDataPoint). Datapoint has methods to set and get value. Developer of application which uses SDT MUST use class dedicated for specific type of SDT Datapoint (E.g. IntegerDataPoint, BooleanDataPoint).

5.6 DataType

5.7 Action

5.8 Event

5.9 Doc

The text for documentation MUST be set in a constructor of the class which implements Doc interface.

6 Data Transfer Objects

RFC 185 defines Data Transfer Objects as a generic means for management solutions to interact with runtime entities in an OSGi Framework. DTOs provides a common, easily serializable representation of the technology.

For all new functionality added to the OSGi Framework the question should be asked: would this feature benefit from a DTO? The expectation is that in most cases it would.

The DTOs for the design in this RFC should be described here and if there are no DTOs being defined an explanation should be given explaining why this is not applicable in this case.

This section is optional and could also be provided in a separate RFC.

7 Javadoc

OSGi Javadoc

28.06.17 02:58

Package Summary		Page
org.osgi.service.sdt	Smart Device Template Package Version 1.0.	18
org.osgi.service.sdt.args	Smart Device Template Package Version 1.0.	44
org.osgi.service.sdt.datapoints	Smart Device Template Package Version 1.0.	58
org.osgi.service.sdt.exceptions	Smart Device Template Package Version 1.0.	76
org.osgi.service.sdt.types	Smart Device Template Package Version 1.0.	80

Package org.osgi.service.sdt

Smart Device Template Package Version 1.0.

See:

Description

Interface Summary		Page
Action		19
Constraint		23
Device		27
Doc		31
Domain		32
Element		34
Event		35
ModuleClass		37
Property		40
SubDevice		41
TypeChoice		43

Class Summary		Page
Arg		21
DataPoint		24

Package org.osgi.service.sdt Description

Smart Device Template Package Version 1.0.

Bundles wishing to use this package must list the package in the Import-Package header of the bundle's manifest. This package has two types of users: the consumers that use the API in this package and the providers that implement the API in this package.

Example import for consumers using the API in this package:

```
Import-Package: org.osgi.service.sdt; version="[1.0,2.0)"
```

Example import for providers implementing the API in this package:

```
Import-Package: org.osgi.service.sdt; version="[1.0,1.1)"
```

Interface Action

org.osgi.service.sdt

All Superinterfaces:

Element

```
public interface Action
extends Element
```

Method Summary		Page
void	**addArg**(Arg arg)	96
Arg	**getArg**(String name)	20
Collection <String>	**getArgNames**()	19
Collection <Arg>	**getArgs**()	19
DataType	**getDataType**()	19
boolean	**isOptional**()	19
void	**removeArg**(String name)	96
void	**setDataType**(DataType type)	19
void	**setOptional**(boolean optional)	19

Methods inherited from interface org.osgi.service.sdt.Element
getDoc, getName, setDoc, setName

Method Detail

getDataType

DataType **getDataType**()

setDataType

void **setDataType**(DataType type)

isOptional

boolean **isOptional**()

setOptional

void **setOptional**(boolean optional)

getArgNames

Collection<String> **getArgNames**()

getArgs

Collection<Arg> **getArgs**()

---

getArg

Arg **getArg**(String name)

addArg

void **addArg**(Arg arg)

removeArg

void **removeArg**(String name)

Class Arg

org.osgi.service.sdt

```
java.lang.Object
  └ org.osgi.service.sdt.Arg
```

All Implemented Interfaces:

Element

Direct Known Subclasses:

ValuedArg

```
public class Arg
extends Object
implements Element
```

Field Summary		Page
Doc	**doc**	21
String	**name**	21
DataType	**type**	21

Constructor Summary	Page
Arg(String name, DataType type)	22

Method Summary		Page
DataType	**getDataType**()	22
Doc	**getDoc**()	22
String	**getName**()	22
void	**setDoc**(Doc doc)	22
void	**setName**(String name)	22

Field Detail

type

DataType **type**

doc

Doc **doc**

name

String **name**

Constructor Detail

Arg

```
public Arg(String name,
           DataType type)
```

Method Detail

getDataType

```
public DataType getDataType()
```

getName

```
public String getName()
```

Specified by:

getName in interface Element

---

setName

```
public void setName(String name)
```

Specified by:

setName in interface Element

---

getDoc

```
public Doc getDoc()
```

Specified by:

getDoc in interface Element

---

setDoc

```
public void setDoc(Doc doc)
```

Specified by:

setDoc in interface Element

Interface Constraint

org.osgi.service.sdt

All Superinterfaces:

Element

```
public interface Constraint
extends Element
```

Method Summary		Page
BasicType	**getType**()	23
String	**getValue**()	23
void	**setType**(BasicType type)	23
void	**setValue**(String value)	23

Methods inherited from interface org.osgi.service.sdt.Element
getDoc, getName, setDoc, setName

Method Detail

getType

BasicType **getType**()

setType

void **setType**(BasicType type)

getValue

String **getValue**()

setValue

void **setValue**(String value)

Class DataPoint

org.osgi.service.sdt

```
java.lang.Object
  └ org.osgi.service.sdt.DataPoint
```

All Implemented Interfaces:

Element

Direct Known Subclasses:

ValuedDataPoint

```
public class DataPoint
extends Object
implements Element
```

Field Summary		Page
Doc	**doc**	25
String	**name**	25
private boolean	**optional**	25
private boolean	**readable**	25
private DataType	**type**	25
private boolean	**writable**	25

Constructor Summary	Page
DataPoint(String name, DataType type)	25

Method Summary		Page
DataType	**getDataType**()	25
Doc	**getDoc**()	26
String	**getName**()	26
boolean	**isOptional**()	25
boolean	**isReadable**()	25
boolean	**isWritable**()	25
void	**setDoc**(Doc doc)	26
void	**setName**(String name)	26
void	**setOptional**(boolean optional)	25
void	**setReadable**(boolean readable)	25
void	**setWritable**(boolean writable)	25

Field Detail

optional

```
private boolean optional
```

readable

```
private boolean readable
```

writable

```
private boolean writable
```

type

```
private DataType type
```

doc

```
Doc doc
```

name

```
String name
```

Constructor Detail

DataPoint

```
public DataPoint(String name,
                 DataType type)
```

Method Detail

getDataType

```
public DataType getDataType()
```

isOptional

```
public boolean isOptional()
```

setOptional

```
public void setOptional(boolean optional)
```

isReadable

```
public boolean isReadable()
```

setReadable

```
public void setReadable(boolean readable)
```

isWritable

```
public boolean isWritable()
```

setWritable

```
public void setWritable(boolean writable)
```

getName

```
public String getName()
```

Specified by:
getName in interface Element

setName

```
public void setName(String name)
```

Specified by:
setName in interface Element

getDoc

```
public Doc getDoc()
```

Specified by:
getDoc in interface Element

setDoc

```
public void setDoc(Doc doc)
```

Specified by:
setDoc in interface Element

Interface Device

org.osgi.service.sdt

All Superinterfaces:

Element

```
public interface Device
extends Element
```

Field Summary		Page
String	**COUNTRY**	29
String	**DEVICE_CATEGORY**	28
String	**DEVICE_LABEL**	28
String	**DEVICE_NAME**	28
String	**DEVICE_TYPE**	28
String	**FW_VERSION**	28
String	**HW_VERSION**	29
String	**LOCATION**	29
String	**MANUCATURING_DATE**	28
String	**MANUFACTURER**	28
String	**MANUFACTURER_DETAILS_LINK**	28
String	**MODEL**	28
String	**OS_VERSION**	29
String	**PRESENTATION_URL**	29
String	**SUB_MODEL**	28
String	**SUPPORT_URL**	29
String	**SW_VERSION**	29
String	**SYSTEM_TIME**	29

Method Summary		Page
void	**addModule**(ModuleClass module)	29
void	**addProperty**(Property property)	30
void	**addSubDevice**(SubDevice device)	30
String	**getInstanceName**()	29
ModuleClass	**getModule**(String name)	29
Collection <String>	**getModuleNames**()	29
Collection <ModuleClass>	**getModules**()	29
Collection <Property>	**getProperties**()	29
Property	**getProperty**(String name)	30

SubDevice	**getSubDevice**(String name)	30
Collection <String>	**getSubDeviceNames**()	30
Collection <SubDevice>	**getSubDevices**()	30
void	**removeModule**(String name)	29
void	**removeProperty**(String name)	30
void	**removeSubDevice**(String name)	30
void	**setInstanceName**()	29
void	**setProperty**(String name, String value)	30
void	**setProperty**(String name, String value, String type)	30

Methods inherited from interface org.osgi.service.sdt.Element
getDoc, getName, setDoc, setName

Field Detail

DEVICE_CATEGORY

```
public static final String DEVICE_CATEGORY = "SDT"
```

DEVICE_LABEL

```
public static final String DEVICE_LABEL = "sdt.device.label"
```

DEVICE_TYPE

```
public static final String DEVICE_TYPE = "sdt.device.type"
```

DEVICE_NAME

```
public static final String DEVICE_NAME = "sdt.device.name"
```

MODEL

```
public static final String MODEL = "sdt.device.model"
```

SUB_MODEL

```
public static final String SUB_MODEL = "sdt.device.submodel"
```

MANUFACTURER

```
public static final String MANUFACTURER = "sdt.device.manufacturer"
```

MANUFACTURER_DETAILS_LINK

```
public static final String MANUFACTURER_DETAILS_LINK = "sdt.device.manufacturer.link"
```

MANUCATURING_DATE

```
public static final String MANUCATURING_DATE = "sdt.device.manufacturer.date"
```

FW_VERSION

```
public static final String FW_VERSION = "sdt.device.version.fw"
```

SW_VERSION

```
public static final String SW_VERSION = "sdt.device.version.sw"
```

HW_VERSION

```
public static final String HW_VERSION = "sdt.device.version.hw"
```

OS_VERSION

```
public static final String OS_VERSION = "sdt.device.version.os"
```

COUNTRY

```
public static final String COUNTRY = "sdt.device.country"
```

LOCATION

```
public static final String LOCATION = "sdt.device.location"
```

SYSTEM_TIME

```
public static final String SYSTEM_TIME = "sdt.device.system.time"
```

SUPPORT_URL

```
public static final String SUPPORT_URL = "sdt.device.support.url"
```

PRESENTATION_URL

```
public static final String PRESENTATION_URL = "sdt.device.presentation.url"
```

Method Detail

getInstanceName

```
String getInstanceName()
```

setInstanceName

```
void setInstanceName()
```

getModuleNames

```
Collection<String> getModuleNames()
```

getModules

```
Collection<ModuleClass> getModules()
```

getModule

```
ModuleClass getModule(String name)
```

addModule

```
void addModule(ModuleClass module)
```

removeModule

```
void removeModule(String name)
```

getProperties

```
Collection<Property> getProperties()
```

getProperty

```
Property getProperty(String name)
```

addProperty

```
void addProperty(Property property)
```

setProperty

```
void setProperty(String name,
                 String value)
```

setProperty

```
void setProperty(String name,
                 String value,
                 String type)
```

removeProperty

```
void removeProperty(String name)
```

getSubDeviceNames

```
Collection<String> getSubDeviceNames()
```

getSubDevices

```
Collection<SubDevice> getSubDevices()
```

getSubDevice

```
SubDevice getSubDevice(String name)
```

addSubDevice

```
void addSubDevice(SubDevice device)
```

removeSubDevice

```
void removeSubDevice(String name)
```

Interface Doc

org.osgi.service.sdt

```
public interface Doc
```

Method Summary		Page
String	getText()	31

Method Detail

getText

```
String getText()
```

Interface Domain

org.osgi.service.sdt

All Superinterfaces:
Element

```
public interface Domain
extends Element
```

Method Summary		Page
void	**addDevice**(Device device)	33
void	**addImport**(Domain domain)	33
void	**addModule**(ModuleClass module)	33
Device	**getDevice**(String name)	33
Collection <String>	**getDeviceNames**()	33
Collection <Device>	**getDevices**()	33
String	**getId**() Note that `getId()` SHALL returns the same value as `getName()` from `Element` interface	32
Domain	**getImport**(String name)	33
Collection <String>	**getImportNames**()	33
Collection <Domain>	**getImports**()	33
ModuleClass	**getModule**(String name)	33
Collection <String>	**getModuleNames**()	33
Collection <ModuleClass>	**getModules**()	33
void	**removeDevice**(String name)	33
void	**removeImport**(String name)	33
void	**removeModule**(String name)	33
void	**setId**(String id)	33

Methods inherited from interface org.osgi.service.sdt.Element
getDoc, getName, setDoc, setName

Method Detail

getId

```
String getId()
```

Note that `getId()` SHALL returns the same value as `getName()` from `Element` interface

setId

```
void setId(String id)
```

getModuleNames

```
Collection<String> getModuleNames()
```

getModules

```
Collection<ModuleClass> getModules()
```

getModule

```
ModuleClass getModule(String name)
```

addModule

```
void addModule(ModuleClass module)
```

removeModule

```
void removeModule(String name)
```

getDeviceNames

```
Collection<String> getDeviceNames()
```

getDevices

```
Collection<Device> getDevices()
```

getDevice

```
Device getDevice(String name)
```

addDevice

```
void addDevice(Device device)
```

removeDevice

```
void removeDevice(String name)
```

getImportNames

```
Collection<String> getImportNames()
```

getImports

```
Collection<Domain> getImports()
```

getImport

```
Domain getImport(String name)
```

addImport

```
void addImport(Domain domain)
```

removeImport

```
void removeImport(String name)
```

Interface Element

org.osgi.service.sdt

All Known Subinterfaces:

Action, Constraint, Device, Domain, ModuleClass

All Known Implementing Classes:

AbstractDateArg, AbstractDateDataPoint, Arg, ArrayDataPoint, BlobArg, BlobDataPoint, BooleanArg, BooleanDataPoint, ByteArg, ByteDataPoint, DataPoint, DataType, DateArg, DateDataPoint, DateTimeArg, DateTimeDataPoint, EnumDataPoint, FloatArg, FloatDataPoint, IntegerArg, IntegerDataPoint, StringArg, StringDataPoint, TimeArg, TimeDataPoint, UriDataPoint, ValuedArg, ValuedDataPoint

---

```
public interface Element
```

Method Summary		Page
Doc	**getDoc**()	34
String	**getName**()	34
void	**setDoc**(Doc doc)	34
void	**setName**(String name)	34

Method Detail

getName

```
String getName()
```

setName

```
void setName(String name)
```

getDoc

```
Doc getDoc()
```

setDoc

```
void setDoc(Doc doc)
```

Interface Event

org.osgi.service.sdt

---

public interface **Event**

Method Summary		Page
void	**addDataPoint**(DataPoint dataPoint)	35
DataPoint	**getDataPoint**(String name)	35
Collection <String>	**getDataPointNames**()	35
Collection <DataPoint>	**getDataPoints**()	35
Date	**getTimeStamp**()	35
Object	**getValue**()	36
boolean	**isOptional**()	20
void	**removeDataPoint**(String name)	35
void	**setOptional**(boolean optional)	35
void	**setValue**(Object value)	36

Method Detail

getTimeStamp

Date **getTimeStamp**()

isOptional

boolean **isOptional**()

setOptional

void **setOptional**(boolean optional)

getDataPointNames

Collection<String> **getDataPointNames**()

getDataPoints

Collection<DataPoint> **getDataPoints**()

getDataPoint

DataPoint **getDataPoint**(String name)

addDataPoint

void **addDataPoint**(DataPoint dataPoint)

removeDataPoint

void **removeDataPoint**(String name)

getValue

```
Object getValue()
```

setValue

```
void setValue(Object value)
```

Interface ModuleClass

org.osgi.service.sdt

All Superinterfaces:
Element

```
public interface ModuleClass
extends Element
```

Method Summary		Page
void	**addAction**(Action action)	38
void	**addDataPoint**(DataPoint dp)	38
void	**addEvent**(Event evt)	39
void	**addProperty**(Property arg)	39
Action	**getAction**(String name)	38
Collection <String>	**getActionNames**()	38
Collection <Action>	**getActions**()	38
DataPoint	**getDataPoint**(String name)	38
Collection <String>	**getDataPointNames**()	38
Collection <DataPoint>	**getDataPoints**()	38
Event	**getEvent**(String name)	39
Collection <String>	**getEventNames**()	38
Collection <Event>	**getEvents**()	38
String	**getInstanceName**()	20
Collection <Property>	**getProperties**()	39
Property	**getProperty**(String name)	39
Collection <String>	**getPropertyNames**()	39
boolean	**isOptional**()	38
void	**removeAction**(String name)	38
void	**removeDataPoint**(String name)	38
void	**removeEvent**(String name)	39
void	**removeProperty**(String name)	39
void	**setInstanceName**(String instanceName)	38
void	**setOptional**(boolean optional)	38
void	**setProperty**(String name, String value)	39

Methods inherited from interface org.osgi.service.sdt.Element
getDoc, getName, setDoc, setName

Method Detail

getInstanceName

String **getInstanceName**()

setInstanceName

void **setInstanceName**(String instanceName)

isOptional

boolean **isOptional**()

setOptional

void **setOptional**(boolean optional)

getActionNames

Collection<String> **getActionNames**()

getActions

Collection<Action> **getActions**()

getAction

Action **getAction**(String name)

addAction

void **addAction**(Action action)

removeAction

void **removeAction**(String name)

getDataPointNames

Collection<String> **getDataPointNames**()

getDataPoints

Collection<DataPoint> **getDataPoints**()

getDataPoint

DataPoint **getDataPoint**(String name)

addDataPoint

void **addDataPoint**(DataPoint dp)

removeDataPoint

void **removeDataPoint**(String name)

getEventNames

Collection<String> **getEventNames**()

getEvents

Collection<Event> **getEvents**()

getEvent

```
Event getEvent(String name)
```

addEvent

```
void addEvent(Event evt)
```

removeEvent

```
void removeEvent(String name)
```

getPropertyNames

```
Collection<String> getPropertyNames()
```

getProperties

```
Collection<Property> getProperties()
```

getProperty

```
Property getProperty(String name)
```

addProperty

```
void addProperty(Property arg)
```

setProperty

```
void setProperty(String name,
                 String value)
```

removeProperty

```
void removeProperty(String name)
```

Interface Property

org.osgi.service.sdt

public interface **Property**

Method Summary		Page
String	**getName**()	40
SimpleType	**getType**()	40
String	**getValue**()	40
boolean	**isOptional**()	40
void	**setOptional**(boolean optional)	40
void	**setType**(SimpleType type)	40
void	**setValue**(String value)	40

Method Detail

getName

String **getName**()

getType

SimpleType **getType**()

setType

void **setType**(SimpleType type)

isOptional

boolean **isOptional**()

setOptional

void **setOptional**(boolean optional)

getValue

String **getValue**()

setValue

void **setValue**(String value)

Interface SubDevice

org.osgi.service.sdt

---

public interface **SubDevice**

Method Summary		Page
void	**addModule**(ModuleClass module)	41
void	**addProperty**(Property property)	42
String	**getInstanceName**()	41
ModuleClass	**getModule**(String name)	41
Collection <String>	**getModuleNames**()	41
Collection <ModuleClass>	**getModules**()	41
Collection <Property>	**getProperties**()	42
Property	**getProperty**(String name)	42
Collection <String>	**getPropertyNames**()	42
void	**removeModule**(String name)	42
void	**removeProperty**(String name)	42
void	**setInstanceName**()	41

Method Detail

getInstanceName

String **getInstanceName**()

setInstanceName

void **setInstanceName**()

getModuleNames

Collection<String> **getModuleNames**()

getModules

Collection<ModuleClass> **getModules**()

getModule

ModuleClass **getModule**(String name)

addModule

void **addModule**(ModuleClass module)

---

removeModule

```
void removeModule(String name)
```

getPropertyNames

```
Collection<String> getPropertyNames()
```

getProperties

```
Collection<Property> getProperties()
```

getProperty

```
Property getProperty(String name)
```

addProperty

```
void addProperty(Property property)
```

removeProperty

```
void removeProperty(String name)
```

Interface TypeChoice

org.osgi.service.sdt

All Known Implementing Classes:
Array, SimpleType, Struct

```
public interface TypeChoice
```

Method Summary		Page
String	getOneM2MType()	43
void	getTypeChoice()	43

Method Detail

getOneM2MType

```
String getOneM2MType()
```

getTypeChoice

```
void getTypeChoice()
```

Package org.osgi.service.sdt.args

Smart Device Template Package Version 1.0.

See:

Description

Class Summary		Page
AbstractDateArg		45
BlobArg		47
BooleanArg		48
ByteArg		49
DateArg		50
DateTimeArg		51
FloatArg		52
IntegerArg		53
StringArg		54
TimeArg		55
ValuedArg		56

Package org.osgi.service.sdt.args Description

Smart Device Template Package Version 1.0.

Bundles wishing to use this package must list the package in the Import-Package header of the bundle's manifest. This package has two types of users: the consumers that use the API in this package and the providers that implement the API in this package.

Example import for consumers using the API in this package:

```
Import-Package: org.osgi.service.sdt.args; version="[1.0,2.0)"
```

Example import for providers implementing the API in this package:

```
Import-Package: org.osgi.service.sdt.args; version="[1.0,1.1)"
```

Class AbstractDateArg

org.osgi.service.sdt.args

```
java.lang.Object
  └ org.osgi.service.sdt.Arg
      └ org.osgi.service.sdt.args.ValuedArg<Date>
          └ org.osgi.service.sdt.args.AbstractDateArg
```

All Implemented Interfaces:

Element

Direct Known Subclasses:

DateArg, DateTimeArg, TimeArg

```
abstract public class AbstractDateArg
extends ValuedArg<Date>
```

Field Summary		Page
private static DateFormat	**dateFormat**	46
private static DateFormat	**dateTimeFormat**	46
private DateFormat	**df**	46
private static DateFormat	**timeFormat**	46

Constructor Summary		Page
protected	**AbstractDateArg**(String name, DataType type)	46

Method Summary		Page
String	**getStringValue**()	46
void	**setValue**(String v)	46
void	**setValue**(long value)	46

Methods inherited from class org.osgi.service.sdt.args.ValuedArg
getValue, setValue

Methods inherited from class org.osgi.service.sdt.Arg
getDataType, getDoc, getName, setDoc, setName

Field Detail

dateFormat

```
private static final DateFormat dateFormat
```

dateTimeFormat

```
private static final DateFormat dateTimeFormat
```

timeFormat

```
private static final DateFormat timeFormat
```

df

```
private DateFormat df
```

Constructor Detail

AbstractDateArg

```
protected AbstractDateArg(String name,
                          DataType type)
```

Method Detail

setValue

```
public void setValue(String v)
```

setValue

```
public void setValue(long value)
```

getStringValue

```
public String getStringValue()
```

Class BlobArg

org.osgi.service.sdt.args

```
java.lang.Object
  └ org.osgi.service.sdt.Arg
      └ org.osgi.service.sdt.args.ValuedArg<byte>
          └ org.osgi.service.sdt.args.BlobArg
```

All Implemented Interfaces:

Element

```
public class BlobArg
extends ValuedArg<byte>
```

Constructor Summary	Page
BlobArg(String name)	47

Methods inherited from class org.osgi.service.sdt.args.ValuedArg
getValue, setValue

Methods inherited from class org.osgi.service.sdt.Arg
getDataType, getDoc, getName, setDoc, setName

Constructor Detail

BlobArg

```
public BlobArg(String name)
```

Class BooleanArg

org.osgi.service.sdt.args

```
java.lang.Object
  └ org.osgi.service.sdt.Arg
      └ org.osgi.service.sdt.args.ValuedArg<Boolean>
          └ org.osgi.service.sdt.args.BooleanArg
```

All Implemented Interfaces:

Element

```
public class BooleanArg
extends ValuedArg<Boolean>
```

Constructor Summary	Page
BooleanArg(String name)	48

Methods inherited from class org.osgi.service.sdt.args.ValuedArg
getValue, setValue

Methods inherited from class org.osgi.service.sdt.Arg
getDataType, getDoc, getName, setDoc, setName

Constructor Detail

BooleanArg

```
public BooleanArg(String name)
```

Class ByteArg

org.osgi.service.sdt.args

```
java.lang.Object
  └ org.osgi.service.sdt.Arg
      └ org.osgi.service.sdt.args.ValuedArg<Byte>
          └ org.osgi.service.sdt.args.ByteArg
```

All Implemented Interfaces:

Element

```
public class ByteArg
extends ValuedArg<Byte>
```

Constructor Summary	Page
ByteArg(String name)	49

Methods inherited from class org.osgi.service.sdt.args.ValuedArg
getValue, setValue

Methods inherited from class org.osgi.service.sdt.Arg
getDataType, getDoc, getName, setDoc, setName

Constructor Detail

ByteArg

```
public ByteArg(String name)
```

Class DateArg

org.osgi.service.sdt.args

```
java.lang.Object
  └ org.osgi.service.sdt.Arg
      └ org.osgi.service.sdt.args.ValuedArg<Date>
          └ org.osgi.service.sdt.args.AbstractDateArg
              └ org.osgi.service.sdt.args.DateArg
```

All Implemented Interfaces:

Element

```
public class DateArg
extends AbstractDateArg
```

Constructor Summary	Page
DateArg(String name)	50

Methods inherited from class org.osgi.service.sdt.args.AbstractDateArg
getStringValue, setValue, setValue

Methods inherited from class org.osgi.service.sdt.args.ValuedArg
getValue, setValue

Methods inherited from class org.osgi.service.sdt.Arg
getDataType, getDoc, getName, setDoc, setName

Constructor Detail

DateArg

```
public DateArg(String name)
```

Class DateTimeArg

org.osgi.service.sdt.args

```
java.lang.Object
  └ org.osgi.service.sdt.Arg
      └ org.osgi.service.sdt.args.ValuedArg<Date>
          └ org.osgi.service.sdt.args.AbstractDateArg
              └ org.osgi.service.sdt.args.DateTimeArg
```

All Implemented Interfaces:

Element

```
public class DateTimeArg
extends AbstractDateArg
```

Constructor Summary	Page
DateTimeArg(String name)	51

Methods inherited from class org.osgi.service.sdt.args.AbstractDateArg
getStringValue, setValue, setValue

Methods inherited from class org.osgi.service.sdt.args.ValuedArg
getValue, setValue

Methods inherited from class org.osgi.service.sdt.Arg
getDataType, getDoc, getName, setDoc, setName

Constructor Detail

DateTimeArg

```
public DateTimeArg(String name)
```

Class FloatArg

org.osgi.service.sdt.args

```
java.lang.Object
  └ org.osgi.service.sdt.Arg
      └ org.osgi.service.sdt.args.ValuedArg<Float>
          └ org.osgi.service.sdt.args.FloatArg
```

All Implemented Interfaces:

Element

```
public class FloatArg
extends ValuedArg<Float>
```

Constructor Summary	Page
FloatArg(String name)	52

Methods inherited from class org.osgi.service.sdt.args.ValuedArg
getValue, setValue

Methods inherited from class org.osgi.service.sdt.Arg
getDataType, getDoc, getName, setDoc, setName

Constructor Detail

FloatArg

```
public FloatArg(String name)
```

Class IntegerArg

org.osgi.service.sdt.args

```
java.lang.Object
  └ org.osgi.service.sdt.Arg
      └ org.osgi.service.sdt.args.ValuedArg<Integer>
          └ org.osgi.service.sdt.args.IntegerArg
```

All Implemented Interfaces:

Element

```
public class IntegerArg
extends ValuedArg<Integer>
```

Constructor Summary	Page
IntegerArg(String name)	53

Methods inherited from class org.osgi.service.sdt.args.ValuedArg
getValue, setValue

Methods inherited from class org.osgi.service.sdt.Arg
getDataType, getDoc, getName, setDoc, setName

Constructor Detail

IntegerArg

```
public IntegerArg(String name)
```

Class StringArg

org.osgi.service.sdt.args

```
java.lang.Object
  └ org.osgi.service.sdt.Arg
      └ org.osgi.service.sdt.args.ValuedArg<String>
          └ org.osgi.service.sdt.args.StringArg
```

All Implemented Interfaces:

Element

```
public class StringArg
extends ValuedArg<String>
```

Constructor Summary	Page
StringArg(String name)	54

Methods inherited from class org.osgi.service.sdt.args.ValuedArg
getValue, setValue

Methods inherited from class org.osgi.service.sdt.Arg
getDataType, getDoc, getName, setDoc, setName

Constructor Detail

StringArg

```
public StringArg(String name)
```

Class TimeArg

org.osgi.service.sdt.args

```
java.lang.Object
  └ org.osgi.service.sdt.Arg
      └ org.osgi.service.sdt.args.ValuedArg<Date>
          └ org.osgi.service.sdt.args.AbstractDateArg
              └ org.osgi.service.sdt.args.TimeArg
```

All Implemented Interfaces:

Element

```
public class TimeArg
extends AbstractDateArg
```

Constructor Summary	Page
TimeArg(String name)	55

Methods inherited from class org.osgi.service.sdt.args.AbstractDateArg
getStringValue, setValue, setValue

Methods inherited from class org.osgi.service.sdt.args.ValuedArg
getValue, setValue

Methods inherited from class org.osgi.service.sdt.Arg
getDataType, getDoc, getName, setDoc, setName

Constructor Detail

TimeArg

```
public TimeArg(String name)
```

Class ValuedArg

org.osgi.service.sdt.args

```
java.lang.Object
  └ org.osgi.service.sdt.Arg
      └ org.osgi.service.sdt.args.ValuedArg
```

All Implemented Interfaces:

Element

Direct Known Subclasses:

AbstractDateArg, BlobArg, BooleanArg, ByteArg, FloatArg, IntegerArg, StringArg

---

```
public class ValuedArg
extends Arg
```

Field Summary		Page
private T	**value**	56

Constructor Summary	Page
ValuedArg(String name, DataType type)	56

Method Summary		Page
T	**getValue**()	56
void	**setValue**(T value)	57

Methods inherited from class org.osgi.service.sdt.Arg
getDataType, getDoc, getName, setDoc, setName

Field Detail

value

```
private T value
```

Constructor Detail

ValuedArg

```
public ValuedArg(String name,
                 DataType type)
```

Method Detail

getValue

```
public T getValue()
```

setValue

```
public void setValue(T value)
```

Package org.osgi.service.sdt.datapoints

Smart Device Template Package Version 1.0.

See:

Description

Class Summary		Page
AbstractDateDataPoint		59
ArrayDataPoint		61
BlobDataPoint		62
BooleanDataPoint		63
ByteDataPoint		64
DateDataPoint		65
DateTimeDataPoint		66
EnumDataPoint		67
FloatDataPoint		69
IntegerDataPoint		70
StringDataPoint		71
TimeDataPoint		72
UriDataPoint		73
ValuedDataPoint		74

Package org.osgi.service.sdt.datapoints Description

Smart Device Template Package Version 1.0.

Bundles wishing to use this package must list the package in the Import-Package header of the bundle's manifest. This package has two types of users: the consumers that use the API in this package and the providers that implement the API in this package.

Example import for consumers using the API in this package:

```
Import-Package: org.osgi.service.sdt.datapoints; version="[1.0,2.0)"
```

Example import for providers implementing the API in this package:

```
Import-Package: org.osgi.service.sdt.datapoints; version="[1.0,1.1)"
```

Class AbstractDateDataPoint

org.osgi.service.sdt.datapoints

```
java.lang.Object
  └ org.osgi.service.sdt.DataPoint
      └ org.osgi.service.sdt.datapoints.ValuedDataPoint<Date>
          └ org.osgi.service.sdt.datapoints.AbstractDateDataPoint
```

All Implemented Interfaces:

Element

Direct Known Subclasses:

DateDataPoint, DateTimeDataPoint, TimeDataPoint

```
abstract public class AbstractDateDataPoint
extends ValuedDataPoint<Date>
```

Field Summary		Page
private static DateFormat	**dateFormat**	60
private static DateFormat	**dateTimeFormat**	60
private DateFormat	**df**	60
private static DateFormat	**timeFormat**	60

Constructor Summary		Page
protected	**AbstractDateDataPoint**(String name, DataType type)	60

Method Summary		Page
String	**getStringValue**()	60
void	**setValue**(String v)	60
void	**setValue**(long value)	60

Methods inherited from class org.osgi.service.sdt.datapoints.ValuedDataPoint
doGetValue, doSetValue, getValue, setValue

Methods inherited from class org.osgi.service.sdt.DataPoint
getDataType, getDoc, getName, isOptional, isReadable, isWritable, setDoc, setName, setOptional, setReadable, setWritable

Field Detail

dateTimeFormat

```
private static final DateFormat dateTimeFormat
```

dateFormat

```
private static final DateFormat dateFormat
```

timeFormat

```
private static final DateFormat timeFormat
```

df

```
private DateFormat df
```

Constructor Detail

AbstractDateDataPoint

```
protected AbstractDateDataPoint(String name,
                                DataType type)
```

Method Detail

setValue

```
public void setValue(String v)
              throws DataPointException,
                     AccessException
```

Throws:

DataPointException
AccessException

---

setValue

```
public void setValue(long value)
              throws DataPointException,
                     AccessException
```

Throws:

DataPointException
AccessException

---

getStringValue

```
public String getStringValue()
                      throws DataPointException,
                             AccessException
```

Throws:

DataPointException
AccessException

Class ArrayDataPoint

org.osgi.service.sdt.datapoints

```
java.lang.Object
  └ org.osgi.service.sdt.DataPoint
      └ org.osgi.service.sdt.datapoints.ValuedDataPoint<List<T>>
          └ org.osgi.service.sdt.datapoints.ArrayDataPoint
```

All Implemented Interfaces:

Element

```
abstract public class ArrayDataPoint
extends ValuedDataPoint<List<T>>
```

Constructor Summary	Page
ArrayDataPoint(String name)	61

Methods inherited from class org.osgi.service.sdt.datapoints.ValuedDataPoint
doGetValue, doSetValue, getValue, setValue

Methods inherited from class org.osgi.service.sdt.DataPoint
getDataType, getDoc, getName, isOptional, isReadable, isWritable, setDoc, setName, setOptional, setReadable, setWritable

Constructor Detail

ArrayDataPoint

```
public ArrayDataPoint(String name)
```

Class BlobDataPoint

org.osgi.service.sdt.datapoints

```
java.lang.Object
  └ org.osgi.service.sdt.DataPoint
      └ org.osgi.service.sdt.datapoints.ValuedDataPoint<byte>
          └ org.osgi.service.sdt.datapoints.BlobDataPoint
```

All Implemented Interfaces:

Element

```
abstract public class BlobDataPoint
extends ValuedDataPoint<byte>
```

Constructor Summary	Page
BlobDataPoint(String name)	62

Methods inherited from class org.osgi.service.sdt.datapoints.ValuedDataPoint
doGetValue, doSetValue, getValue, setValue

Methods inherited from class org.osgi.service.sdt.DataPoint
getDataType, getDoc, getName, isOptional, isReadable, isWritable, setDoc, setName, setOptional, setReadable, setWritable

Constructor Detail

BlobDataPoint

```
public BlobDataPoint(String name)
```

Class BooleanDataPoint

org.osgi.service.sdt.datapoints

```
java.lang.Object
  └ org.osgi.service.sdt.DataPoint
      └ org.osgi.service.sdt.datapoints.ValuedDataPoint<Boolean>
          └ org.osgi.service.sdt.datapoints.BooleanDataPoint
```

All Implemented Interfaces:

Element

```
abstract public class BooleanDataPoint
extends ValuedDataPoint<Boolean>
```

Constructor Summary	Page
BooleanDataPoint(String name)	63

Methods inherited from class org.osgi.service.sdt.datapoints.ValuedDataPoint
doGetValue, doSetValue, getValue, setValue

Methods inherited from class org.osgi.service.sdt.DataPoint
getDataType, getDoc, getName, isOptional, isReadable, isWritable, setDoc, setName, setOptional, setReadable, setWritable

Constructor Detail

BooleanDataPoint

```
public BooleanDataPoint(String name)
```

Class ByteDataPoint

org.osgi.service.sdt.datapoints

```
java.lang.Object
  └ org.osgi.service.sdt.DataPoint
      └ org.osgi.service.sdt.datapoints.ValuedDataPoint<Byte>
          └ org.osgi.service.sdt.datapoints.ByteDataPoint
```

All Implemented Interfaces:

Element

```
abstract public class ByteDataPoint
extends ValuedDataPoint<Byte>
```

Constructor Summary	*Page*
ByteDataPoint(String name)	64

Methods inherited from class org.osgi.service.sdt.datapoints.ValuedDataPoint
doGetValue, doSetValue, getValue, setValue

Methods inherited from class org.osgi.service.sdt.DataPoint
getDataType, getDoc, getName, isOptional, isReadable, isWritable, setDoc, setName, setOptional, setReadable, setWritable

Constructor Detail

ByteDataPoint

```
public ByteDataPoint(String name)
```

Class DateDataPoint

org.osgi.service.sdt.datapoints

```
java.lang.Object
  └ org.osgi.service.sdt.DataPoint
      └ org.osgi.service.sdt.datapoints.ValuedDataPoint<Date>
          └ org.osgi.service.sdt.datapoints.AbstractDateDataPoint
              └ org.osgi.service.sdt.datapoints.DateDataPoint
```

All Implemented Interfaces:

Element

```
abstract public class DateDataPoint
extends AbstractDateDataPoint
```

Constructor Summary	Page
DateDataPoint(String name)	65

Methods inherited from class org.osgi.service.sdt.datapoints.AbstractDateDataPoint
getStringValue, setValue, setValue

Methods inherited from class org.osgi.service.sdt.datapoints.ValuedDataPoint
doGetValue, doSetValue, getValue, setValue

Methods inherited from class org.osgi.service.sdt.DataPoint
getDataType, getDoc, getName, isOptional, isReadable, isWritable, setDoc, setName, setOptional, setReadable, setWritable

Constructor Detail

DateDataPoint

```
public DateDataPoint(String name)
```

Class DateTimeDataPoint

org.osgi.service.sdt.datapoints

```
java.lang.Object
  └ org.osgi.service.sdt.DataPoint
      └ org.osgi.service.sdt.datapoints.ValuedDataPoint<Date>
          └ org.osgi.service.sdt.datapoints.AbstractDateDataPoint
              └ org.osgi.service.sdt.datapoints.DateTimeDataPoint
```

All Implemented Interfaces:

Element

```
abstract public class DateTimeDataPoint
extends AbstractDateDataPoint
```

Constructor Summary	Page
DateTimeDataPoint(String name)	66

Methods inherited from class org.osgi.service.sdt.datapoints.AbstractDateDataPoint
getStringValue, setValue, setValue

Methods inherited from class org.osgi.service.sdt.datapoints.ValuedDataPoint
doGetValue, doSetValue, getValue, setValue

Methods inherited from class org.osgi.service.sdt.DataPoint
getDataType, getDoc, getName, isOptional, isReadable, isWritable, setDoc, setName, setOptional, setReadable, setWritable

Constructor Detail

DateTimeDataPoint

```
public DateTimeDataPoint(String name)
```

Class EnumDataPoint

org.osgi.service.sdt.datapoints

```
java.lang.Object
  └ org.osgi.service.sdt.DataPoint
      └ org.osgi.service.sdt.datapoints.ValuedDataPoint<T>
          └ org.osgi.service.sdt.datapoints.EnumDataPoint
```

All Implemented Interfaces:

Element

```
abstract public class EnumDataPoint
extends ValuedDataPoint<T>
```

Field Summary		Page
private List<T>	**values**	67

Constructor Summary	Page
EnumDataPoint(String name)	68
EnumDataPoint(String name, DataType type)	68

Method Summary		Page
List<T>	**getValidValues**()	68
void	**setValidValues**(Collection<T> v)	68
void	**setValidValues**(T[] v)	68
void	**setValue**(T v)	68

Methods inherited from class org.osgi.service.sdt.datapoints.ValuedDataPoint
doGetValue, doSetValue, getValue

Methods inherited from class org.osgi.service.sdt.DataPoint
getDataType, getDoc, getName, isOptional, isReadable, isWritable, setDoc, setName, setOptional, setReadable, setWritable

Field Detail

values

```
private List<T> values
```

Constructor Detail

EnumDataPoint

```
public EnumDataPoint(String name)
```

EnumDataPoint

```
public EnumDataPoint(String name,
                     DataType type)
```

Method Detail

setValue

```
public void setValue(T v)
              throws DataPointException,
                     AccessException
```

Overrides:

setValue in class ValuedDataPoint

Throws:

DataPointException
AccessException

setValidValues

```
public void setValidValues(T[] v)
```

setValidValues

```
public void setValidValues(Collection<T> v)
```

getValidValues

```
public List<T> getValidValues()
```

Class FloatDataPoint

org.osgi.service.sdt.datapoints

```
java.lang.Object
  └ org.osgi.service.sdt.DataPoint
      └ org.osgi.service.sdt.datapoints.ValuedDataPoint<Float>
          └ org.osgi.service.sdt.datapoints.FloatDataPoint
```

All Implemented Interfaces:

Element

```
abstract public class FloatDataPoint
extends ValuedDataPoint<Float>
```

Constructor Summary	Page
FloatDataPoint(String name)	69

Methods inherited from class org.osgi.service.sdt.datapoints.ValuedDataPoint
doGetValue, doSetValue, getValue, setValue

Methods inherited from class org.osgi.service.sdt.DataPoint
getDataType, getDoc, getName, isOptional, isReadable, isWritable, setDoc, setName, setOptional, setReadable, setWritable

Constructor Detail

FloatDataPoint

```
public FloatDataPoint(String name)
```

Class IntegerDataPoint

org.osgi.service.sdt.datapoints

```
java.lang.Object
  └ org.osgi.service.sdt.DataPoint
      └ org.osgi.service.sdt.datapoints.ValuedDataPoint<Integer>
          └ org.osgi.service.sdt.datapoints.IntegerDataPoint
```

All Implemented Interfaces:

Element

```
abstract public class IntegerDataPoint
extends ValuedDataPoint<Integer>
```

Constructor Summary	Page
IntegerDataPoint(String name)	70

Methods inherited from class org.osgi.service.sdt.datapoints.ValuedDataPoint
doGetValue, doSetValue, getValue, setValue

Methods inherited from class org.osgi.service.sdt.DataPoint
getDataType, getDoc, getName, isOptional, isReadable, isWritable, setDoc, setName, setOptional, setReadable, setWritable

Constructor Detail

IntegerDataPoint

```
public IntegerDataPoint(String name)
```

Class StringDataPoint

org.osgi.service.sdt.datapoints

```
java.lang.Object
  └ org.osgi.service.sdt.DataPoint
      └ org.osgi.service.sdt.datapoints.ValuedDataPoint<String>
          └ org.osgi.service.sdt.datapoints.StringDataPoint
```

All Implemented Interfaces:

Element

```
abstract public class StringDataPoint
extends ValuedDataPoint<String>
```

Constructor Summary	Page
StringDataPoint(String name)	71

Methods inherited from class org.osgi.service.sdt.datapoints.ValuedDataPoint
doGetValue, doSetValue, getValue, setValue

Methods inherited from class org.osgi.service.sdt.DataPoint
getDataType, getDoc, getName, isOptional, isReadable, isWritable, setDoc, setName, setOptional, setReadable, setWritable

Constructor Detail

StringDataPoint

```
public StringDataPoint(String name)
```

Class TimeDataPoint

org.osgi.service.sdt.datapoints

```
java.lang.Object
  └ org.osgi.service.sdt.DataPoint
      └ org.osgi.service.sdt.datapoints.ValuedDataPoint<Date>
          └ org.osgi.service.sdt.datapoints.AbstractDateDataPoint
              └ org.osgi.service.sdt.datapoints.TimeDataPoint
```

All Implemented Interfaces:

Element

```
abstract public class TimeDataPoint
extends AbstractDateDataPoint
```

Constructor Summary	Page
TimeDataPoint(String name)	72

Methods inherited from class org.osgi.service.sdt.datapoints.AbstractDateDataPoint
getStringValue, setValue, setValue

Methods inherited from class org.osgi.service.sdt.datapoints.ValuedDataPoint
doGetValue, doSetValue, getValue, setValue

Methods inherited from class org.osgi.service.sdt.DataPoint
getDataType, getDoc, getName, isOptional, isReadable, isWritable, setDoc, setName, setOptional, setReadable, setWritable

Constructor Detail

TimeDataPoint

```
public TimeDataPoint(String name)
```

Class UriDataPoint

org.osgi.service.sdt.datapoints

```
java.lang.Object
  └ org.osgi.service.sdt.DataPoint
      └ org.osgi.service.sdt.datapoints.ValuedDataPoint<URI>
          └ org.osgi.service.sdt.datapoints.UriDataPoint
```

All Implemented Interfaces:

Element

```
abstract public class UriDataPoint
extends ValuedDataPoint<URI>
```

Constructor Summary	Page
UriDataPoint(String name)	73

Method Summary		Page
void	**setValue**(String v)	73

Methods inherited from class org.osgi.service.sdt.datapoints.ValuedDataPoint
doGetValue, doSetValue, getValue, setValue

Methods inherited from class org.osgi.service.sdt.DataPoint
getDataType, getDoc, getName, isOptional, isReadable, isWritable, setDoc, setName, setOptional, setReadable, setWritable

Constructor Detail

UriDataPoint

```
public UriDataPoint(String name)
```

Method Detail

setValue

```
public void setValue(String v)
             throws DataPointException,
                    AccessException
```

Throws:

DataPointException
AccessException

Class ValuedDataPoint

org.osgi.service.sdt.datapoints

```
java.lang.Object
  └ org.osgi.service.sdt.DataPoint
      └ org.osgi.service.sdt.datapoints.ValuedDataPoint
```

All Implemented Interfaces:

Element

Direct Known Subclasses:

AbstractDateDataPoint, ArrayDataPoint, BlobDataPoint, BooleanDataPoint, ByteDataPoint, EnumDataPoint, FloatDataPoint, IntegerDataPoint, StringDataPoint, UriDataPoint

---

```
abstract public class ValuedDataPoint
extends DataPoint
```

Constructor Summary	Page
ValuedDataPoint(String name, DataType type)	74

Method Summary		Page
protected abstract T	**doGetValue**()	75
protected abstract void	**doSetValue**(T value)	75
T	**getValue**()	74
void	**setValue**(T value)	75

Methods inherited from class org.osgi.service.sdt.DataPoint
getDataType, getDoc, getName, isOptional, isReadable, isWritable, setDoc, setName, setOptional, setReadable, setWritable

Constructor Detail

ValuedDataPoint

```
public ValuedDataPoint(String name,
                       DataType type)
```

Method Detail

getValue

```
public T getValue()
           throws DataPointException,
                  AccessException
```

Throws:

DataPointException

AccessException

---

setValue

```
public void setValue(T value)
              throws DataPointException,
                     AccessException
```

Throws:

DataPointException
AccessException

---

doGetValue

```
protected abstract T doGetValue()
                          throws DataPointException
```

Throws:

DataPointException

---

doSetValue

```
protected abstract void doSetValue(T value)
                            throws DataPointException
```

Throws:

DataPointException

Package org.osgi.service.sdt.exceptions

Smart Device Template Package Version 1.0.

See:

Description

Exception Summary		Page
AccessException		77
ActionException		78
DataPointException		79

Package org.osgi.service.sdt.exceptions Description

Smart Device Template Package Version 1.0.

Bundles wishing to use this package must list the package in the Import-Package header of the bundle's manifest. This package has two types of users: the consumers that use the API in this package and the providers that implement the API in this package.

Example import for consumers using the API in this package:

```
Import-Package: org.osgi.service.sdt.exceptions; version="[1.0,2.0)"
```

Example import for providers implementing the API in this package:

```
Import-Package: org.osgi.service.sdt.exceptions; version="[1.0,1.1)"
```

Class AccessException

org.osgi.service.sdt.exceptions

```
java.lang.Object
  └java.lang.Throwable
      └java.lang.Exception
          └org.osgi.service.sdt.exceptions.AccessException
```

All Implemented Interfaces:

Serializable

```
public class AccessException
extends Exception
```

Constructor Summary	Page
AccessException()	77
AccessException(String message)	77
AccessException(String message, Throwable cause)	77
AccessException(Throwable cause)	77

Constructor Detail

AccessException

```
public AccessException()
```

~~AccessException~~

```
public AccessException(String message)
```

~~AccessException~~

```
public AccessException(Throwable cause)
```

~~AccessException~~

```
public AccessException(String message,
                       Throwable cause)
```

Class ActionException

org.osgi.service.sdt.exceptions

```
java.lang.Object
  └ java.lang.Throwable
      └ java.lang.Exception
          └ org.osgi.service.sdt.exceptions.ActionException
```

All Implemented Interfaces:
Serializable

```
public class ActionException
extends Exception
```

Constructor Summary	Page
ActionException()	78
ActionException(String message)	78
ActionException(String message, Throwable cause)	78
ActionException(Throwable cause)	78

Constructor Detail

ActionException

```
public ActionException()
```

ActionException

```
public ActionException(String message)
```

ActionException

```
public ActionException(Throwable cause)
```

ActionException

```
public ActionException(String message,
                       Throwable cause)
```

Class DataPointException

org.osgi.service.sdt.exceptions

```
java.lang.Object
  └ java.lang.Throwable
      └ java.lang.Exception
          └ org.osgi.service.sdt.exceptions.DataPointException
```

All Implemented Interfaces:
Serializable

```
public class DataPointException
extends Exception
```

Constructor Summary	Page
DataPointException()	79
DataPointException(String message)	79
DataPointException(String message, Throwable cause)	79
DataPointException(Throwable cause)	79

Constructor Detail

DataPointException

```
public DataPointException()
```

DataPointException

```
public DataPointException(String message)
```

DataPointException

```
public DataPointException(Throwable cause)
```

DataPointException

```
public DataPointException(String message,
                          Throwable cause)
```

Package org.osgi.service.sdt.types

Smart Device Template Package Version 1.0.

See:

Description

Class Summary		Page
Array		81
BasicType		83
DataType		86
SimpleType		90
Struct	Abstract class - new class extending this Struct class will be needed in future	93

Package org.osgi.service.sdt.types Description

Smart Device Template Package Version 1.0.

Bundles wishing to use this package must list the package in the Import-Package header of the bundle's manifest. This package has two types of users: the consumers that use the API in this package and the providers that implement the API in this package.

Example import for consumers using the API in this package:

```
Import-Package: org.osgi.service.sdt.types; version="[1.0,2.0)"
```

Example import for providers implementing the API in this package:

```
Import-Package: org.osgi.service.sdt.types; version="[1.0,1.1)"
```

Class Array

org.osgi.service.sdt.types

```
java.lang.Object
  └ org.osgi.service.sdt.types.Array
```

All Implemented Interfaces:

TypeChoice

```
public class Array
extends Object
implements TypeChoice
```

Field Summary		Page
private ArrayList<DataType>	**dataTypes**	81

Constructor Summary	Page
Array()	81

Method Summary		Page
void	**addDataType**(DataType dataType)	81
ArrayList<DataType>	**getDataTypes**()	81
String	**getOneM2MType**()	82
void	**getTypeChoice**()	82
void	**removeDataType**(DataType dataType)	82

Field Detail

dataTypes

```
private ArrayList<DataType> dataTypes
```

Constructor Detail

Array

```
public Array()
```

Method Detail

getDataTypes

```
public ArrayList<DataType> getDataTypes()
```

addDataType

```
public void addDataType(DataType dataType)
```

removeDataType

```
public void removeDataType(DataType dataType)
```

getOneM2MType

```
public String getOneM2MType()
```

Specified by:

getOneM2MType in interface TypeChoice

getTypeChoice

```
public void getTypeChoice()
```

Specified by:

getTypeChoice in interface TypeChoice

Class BasicType

org.osgi.service.sdt.types

```
java.lang.Object
  └ org.osgi.service.sdt.types.BasicType
```

```
public class BasicType
extends Object
```

Field Summary		Page
static BasicType	**BLOB**	84
static BasicType	**BOOLEAN**	84
static BasicType	**BYTE**	84
private Class<?>	**clazz**	84
static BasicType	**DATE**	84
static BasicType	**DATETIME**	84
static BasicType	**ENUM**	84
static BasicType	**FLOAT**	84
static BasicType	**INTEGER**	84
static BasicType	**STRING**	84
static BasicType	**TIME**	84
static BasicType	**URI**	84
private String	**value**	84
private static Map<String,BasicType>	**values**	84

Constructor Summary		Page
protected	**BasicType**(String v, Class<?> c)	84

Method Summary		Page
static BasicType	**getBasicType**(String s)	85
Class<?>	**getClazz**()	85
String	**getValue**()	85

Field Detail

values

```
private static Map<String,BasicType> values
```

INTEGER

```
public static final BasicType INTEGER
```

BOOLEAN

```
public static final BasicType BOOLEAN
```

STRING

```
public static final BasicType STRING
```

BYTE

```
public static final BasicType BYTE
```

FLOAT

```
public static final BasicType FLOAT
```

ENUM

```
public static final BasicType ENUM
```

DATE

```
public static final BasicType DATE
```

TIME

```
public static final BasicType TIME
```

DATETIME

```
public static final BasicType DATETIME
```

BLOB

```
public static final BasicType BLOB
```

URI

```
public static final BasicType URI
```

value

```
private final String value
```

clazz

```
private final Class<?> clazz
```

Constructor Detail

BasicType

```
protected BasicType(String v,
                    Class<?> c)
```

Method Detail

getValue

```
public String getValue()
```

getClazz

```
public Class<?> getClazz()
```

getBasicType

```
public static BasicType getBasicType(String s)
```

Class DataType

org.osgi.service.sdt.types

java.lang.Object
  └ **org.osgi.service.sdt.types.DataType**

All Implemented Interfaces:
Element

---

public class **DataType**
extends Object
implements Element

Field Summary		Page
static DataType	**Blob**	88
static DataType	**Boolean**	87
static DataType	**Byte**	87
private Map<String,Constraint>	**constraints**	88
static DataType	**Date**	87
static DataType	**Datetime**	87
Doc	**doc**	87
static DataType	**Enum**	87
static DataType	**Float**	87
static DataType	**Integer**	87
String	**name**	87
static DataType	**String**	87
static DataType	**Time**	87
private TypeChoice	**type**	88
private String	**unitOfMeasure**	88
static DataType	**Uri**	88

Constructor Summary	Page
DataType(String name, TypeChoice type)	88

Method Summary		Page
void	**addConstraint**(Constraint constraint)	88

Constraint	**getConstraint**(String name)	88
Collection <String>	**getConstraintNames**()	88
Collection <Constraint>	**getConstraints**()	88
Doc	**getDoc**()	89
String	**getName**()	88
TypeChoice	**getTypeChoice**()	88
String	**getUnitOfMeasure**()	88
void	**removeConstraint**(String name)	88
void	**setDoc**(Doc doc)	89
void	**setName**(String name)	89
void	**setUnitOfMeasure**(String unitOfMeasure)	88

Field Detail

doc

Doc **doc**

~~name~~

String **name**

~~Boolean~~

public static final DataType **Boolean**

~~Byte~~

public static final DataType **Byte**

~~Integer~~

public static final DataType **Integer**

~~Float~~

public static final DataType **Float**

~~String~~

public static final DataType **String**

~~Enum~~

public static final DataType **Enum**

~~Date~~

public static final DataType **Date**

~~Time~~

public static final DataType **Time**

~~Datetime~~

public static final DataType **Datetime**

Blob

public static final DataType **Blob**

Uri

public static final DataType **Uri**

unitOfMeasure

private String **unitOfMeasure**

type

private TypeChoice **type**

constraints

private Map<String,Constraint> **constraints**

Constructor Detail

DataType

public **DataType**(String name,
TypeChoice type)

Method Detail

getTypeChoice

public TypeChoice **getTypeChoice**()

getUnitOfMeasure

public String **getUnitOfMeasure**()

setUnitOfMeasure

public void **setUnitOfMeasure**(String unitOfMeasure)

getConstraintNames

public Collection<String> **getConstraintNames**()

getConstraints

public Collection<Constraint> **getConstraints**()

getConstraint

public Constraint **getConstraint**(String name)

addConstraint

public void **addConstraint**(Constraint constraint)

removeConstraint

public void **removeConstraint**(String name)

getName

public String **getName**()

Specified by:

getName in interface Element

---

getDoc

```
public Doc getDoc()
```

Specified by:

getDoc in interface Element

---

setDoc

```
public void setDoc(Doc doc)
```

Specified by:

setDoc in interface Element

---

setName

```
public void setName(String name)
```

Specified by:

setName in interface Element

Class SimpleType

org.osgi.service.sdt.types

```
java.lang.Object
  └ org.osgi.service.sdt.types.SimpleType
```

All Implemented Interfaces:

TypeChoice

```
public class SimpleType
extends Object
implements TypeChoice
```

Field Summary		Page
static SimpleType	**Blob**	91
static SimpleType	**Boolean**	91
static SimpleType	**Byte**	91
static SimpleType	**Date**	91
static SimpleType	**Datetime**	91
static SimpleType	**Enum**	91
static SimpleType	**Float**	91
static SimpleType	**Integer**	91
static SimpleType	**String**	91
static SimpleType	**Time**	91
private BasicType	**type**	91
static SimpleType	**Uri**	91
private static Map<String,SimpleType>	**values**	91

Constructor Summary		Page
protected	**SimpleType**(BasicType type)	91

Method Summary		Page
String	**getOneM2MType**()	92
static SimpleType	**getSimpleType**(String s)	92
BasicType	**getType**()	91
void	**getTypeChoice**()	92

Field Detail

values

private static Map<String,SimpleType> **values**

~~Boolean~~

public static final SimpleType **Boolean**

~~Byte~~

public static final SimpleType **Byte**

~~Integer~~

public static final SimpleType **Integer**

~~Float~~

public static final SimpleType **Float**

~~String~~

public static final SimpleType **String**

~~Enum~~

public static final SimpleType **Enum**

~~Date~~

public static final SimpleType **Date**

~~Time~~

public static final SimpleType **Time**

~~Datetime~~

public static final SimpleType **Datetime**

~~Blob~~

public static final SimpleType **Blob**

~~Uri~~

public static final SimpleType **Uri**

~~type~~

private BasicType **type**

Constructor Detail

SimpleType

protected **SimpleType**(BasicType type)

Method Detail

getType

public BasicType **getType**()

getSimpleType

```
public static SimpleType getSimpleType(String s)
```

getOneM2MType

```
public String getOneM2MType()
```

Specified by:

getOneM2MType in interface TypeChoice

getTypeChoice

```
public void getTypeChoice()
```

Specified by:

getTypeChoice in interface TypeChoice

Class Struct

org.osgi.service.sdt.types

```
java.lang.Object
  └ org.osgi.service.sdt.types.Struct
```

All Implemented Interfaces:

TypeChoice

```
abstract public class Struct
extends Object
implements TypeChoice
```

Abstract class - new class extending this Struct class will be needed in future

---

Field Summary		Page
private Collection<DataType>	**dataTypes**	93

Constructor Summary	Page
Struct()	93

Method Summary		Page
void	**addDataType**(DataType dataType)	94
Collection<DataType>	**getDataTypes**()	93
void	**removeDataType**(DataType dataType)	94

Methods inherited from interface org.osgi.service.sdt.TypeChoice
getOneM2MType, getTypeChoice

Field Detail

dataTypes

```
private Collection<DataType> dataTypes
```

Constructor Detail

Struct

```
public Struct()
```

Method Detail

getDataTypes

```
public Collection<DataType> getDataTypes()
```

addDataType

```
public void addDataType(DataType dataType)
```

removeDataType

```
public void removeDataType(DataType dataType)
```

---

8 Considered Alternatives

For posterity, record the design alternatives that were considered but rejected along with the reason for rejection. This is especially important for external/earlier solutions that were deemed not applicable.

9 Security Considerations

Description of all known vulnerabilities this may either introduce or address as well as scenarios of how the weaknesses could be circumvented.

10 Document Support

10.1 References

[1] OneM2M, Home Appliances Information Model and Mapping, TS-0023-V2.0.0, 2016
[2]

Add references simply by adding new items. You can then cross-refer to them by chosing <Insert><Cross Reference><Numbered Item> and then selecting the paragraph. ***STATIC REFERENCES (I.E. BODGED) ARE NOT ACCEPTABLE, SOMEONE WILL HAVE TO UPDATE THEM LATER, SO DO IT PROPERLY NOW.***

10.2 Author's Address

Name	Maciej Goluch
Company	Orange Polska S.A.
Address	Obrzezna 7, Warsaw, Poland
e-mail	maciej.goluch@orange.com

Name	Andre Bottaro
Company	Orange Labs
Address	6 avenue Albert Durand, 31700 Toulouse, France
e-mail	andre.bottaro@orange.com

Name	Pawel Strzemecki
Company	Orange Polska S.A.
Address	Obrzezna 7, Warsaw, Poland
e-mail	-

10.3 Acronyms and Abbreviations

10.4 End of Document