INTERNATIONAL STANDARD

ISO/IEC 29341-30-11

First edition
2017-06

# Information technology — UPnP Device Architecture —

## Part 30-11: IoT management and control device control protocol — IoT management and control data model service

*Technologies de l'information — Architecture de dispositif UPnP —*

*Partie 30-11: Protocole de contrôle de dispositif de gestion et de contrôle de l'Internet des objets — Service de modèle de données de gestion et de contrôle de l'Internet des objets*

Reference number
ISO/IEC 29341-30-11:2017(E)

ISO copyright office<br>
Ch. de Blandonnet 8 • CP 401
CH-1214 Vernier, Geneva, Switzerland
Tel. +41 22 749 01 11
Fax +41 22 749 09 47
copyright@iso.org
www.iso.org

## CONTENTS

# Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see http://www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation on the voluntary nature of Standards, the meaning of the ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the WTO principles in the Technical Barriers to Trade (TBT) see the following URL: Foreword – Supplementary information

ISO/IEC 29341-30-11 was prepared by UPnP Forum and adopted, under the PAS procedure, by joint technical committee ISO/IEC JTC 1, *Information technology*, in parallel with its approval by national bodies of ISO and IEC.

The list of all currently available parts of ISO/IEC 29341 series, under the general title *Information technology — UPnP Device Architecture*, can be found on the ISO web site.

# Introduction

ISO and IEC draw attention to the fact that it is claimed that compliance with this document may involve the use of patents as indicated below.

ISO and IEC take no position concerning the evidence, validity and scope of these patent rights. The holders of -these patent rights have assured ISO and IEC that they are willing to negotiate licenses under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statements of the holders of these patent rights are registered with ISO and IEC.

Intel Corporation has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Intel Corporation
Standards Licensing Department
5200 NE Elam Young Parkway
MS: JFS-98
USA – Hillsboro, Oregon 97124

Microsoft Corporation has informed IEC and ISO that it has patent applications or granted patents as listed below:

6101499 / US; 6687755 / US; 6910068 / US; 7130895 / US; 6725281 / US; 7089307 / US; 7069312 / US; 10/783 524 /US

Information may be obtained from:

Microsoft Corporation
One Microsoft Way
USA – Redmond WA 98052

Philips International B.V. has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Philips International B.V. – IP&S
High Tech campus, building 44 3A21
NL – 5656 Eindhoven

NXP B.V. (NL) has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

NXP B.V. (NL)
High Tech campus 60
NL – 5656 AG Eindhoven

Matsushita Electric Industrial Co. Ltd. has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Matsushita Electric Industrial Co. Ltd.
1-3-7 Shiromi, Chuoh-ku
JP – Osaka 540-6139

Hewlett Packard Company has informed IEC and ISO that it has patent applications or granted patents as listed below:

5 956 487 / US; 6 170 007 / US; 6 139 177 / US; 6 529 936 / US; 6 470 339 / US; 6 571 388 / US; 6 205 466 / US

Information may be obtained from:

Hewlett Packard Company
1501 Page Mill Road
USA – Palo Alto, CA 94304

Samsung Electronics Co. Ltd. has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Digital Media Business, Samsung Electronics Co. Ltd.
416 Maetan-3 Dong, Yeongtang-Gu,
KR – Suwon City 443-742

Huawei Technologies Co., Ltd. has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Huawei Technologies Co., Ltd.
Administration Building, Bantian Longgang District
Shenzhen – China 518129

Qualcomm Incorporated has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Qualcomm Incorporated
5775 Morehouse Drive
San Diego, CA – USA 92121

Telecom Italia S.p.A.has informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Telecom Italia S.p.A.
Via Reiss Romoli, 274
Turin - Italy 10148

Cisco Systems informed IEC and ISO that it has patent applications or granted patents.

Information may be obtained from:

Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA – USA 95134

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those identified above. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

**Original UPnP Document**

Reference may be made in this document to original UPnP documents. These references are retained in order to maintain consistency between the specifications as published by ISO/IEC and by UPnP Implementers Corporation and later by UPnP Forum. The following table indicates the original UPnP document titles and the corresponding part of ISO/IEC 29341:

| UPnP Document Title | ISO/IEC 29341 Part |
|---|---|
| UPnP Device Architecture 1.0 | ISO/IEC 29341-1:2008 |
| UPnP Device Architecture Version 1.0 | ISO/IEC 29341-1:2011 |
| UPnP Device Architecture 1.1 | ISO/IEC 29341-1-1:2011 |
| UPnP Device Architecture 2.0 | ISO/IEC 29341-1-2 |
| UPnP Basic:1 Device | ISO/IEC 29341-2 |
| UPnP AV Architecture:1 | ISO/IEC 29341-3-1:2008 |
| UPnP AV Architecture:1 | ISO/IEC 29341-3-1:2011 |
| UPnP AVTransport:1 Service | ISO/IEC 29341-3-10 |
| UPnP ConnectionManager:1 Service | ISO/IEC 29341-3-11 |
| UPnP ContentDirectory:1 Service | ISO/IEC 29341-3-12 |
| UPnP RenderingControl:1 Service | ISO/IEC 29341-3-13 |
| UPnP MediaRenderer:1 Device | ISO/IEC 29341-3-2 |
| UPnP MediaRenderer:2 Device | ISO/IEC 29341-3-2:2011 |
| UPnP MediaServer:1 Device | ISO/IEC 29341-3-3 |
| UPnP AVTransport:2 Service | ISO/IEC 29341-4-10:2008 |
| UPnP AVTransport:2 Service | ISO/IEC 29341-4-10:2011 |
| UPnP ConnectionManager:2 Service | ISO/IEC 29341-4-11:2008 |
| UPnP ConnectionManager:2 Service | ISO/IEC 29341-4-11:2011 |
| UPnP ContentDirectory:2 Service | ISO/IEC 29341-4-12 |
| UPnP RenderingControl:2 Service | ISO/IEC 29341-4-13:2008 |
| UPnP RenderingControl:2 Service | ISO/IEC 29341-4-13:2011 |
| UPnP ScheduledRecording:1 | ISO/IEC 29341-4-14 |
| UPnP ScheduledRecording:2 | ISO/IEC 29341-4-14:2011 |
| UPnP MediaRenderer:2 Device | ISO/IEC 29341-4-2 |
| UPnP MediaServer:2 Device | ISO/IEC 29341-4-3 |
| UPnP AV Datastructure Template:1 | ISO/IEC 29341-4-4:2008 |
| UPnP AV Datastructure Template:1 | ISO/IEC 29341-4-4:2011 |
| UPnP DigitalSecurityCamera:1 Device | ISO/IEC 29341-5-1 |
| UPnP DigitalSecurityCameraMotionImage:1 Service | ISO/IEC 29341-5-10 |
| UPnP DigitalSecurityCameraSettings:1 Service | ISO/IEC 29341-5-11 |
| UPnP DigitalSecurityCameraStillImage:1 Service | ISO/IEC 29341-5-12 |
| UPnP HVAC_System:1 Device | ISO/IEC 29341-6-1 |
| UPnP ControlValve:1 Service | ISO/IEC 29341-6-10 |
| UPnP HVAC_FanOperatingMode:1 Service | ISO/IEC 29341-6-11 |
| UPnP FanSpeed:1 Service | ISO/IEC 29341-6-12 |
| UPnP HouseStatus:1 Service | ISO/IEC 29341-6-13 |
| UPnP HVAC_SetpointSchedule:1 Service | ISO/IEC 29341-6-14 |
| UPnP TemperatureSensor:1 Service | ISO/IEC 29341-6-15 |
| UPnP TemperatureSetpoint:1 Service | ISO/IEC 29341-6-16 |
| UPnP HVAC_UserOperatingMode:1 Service | ISO/IEC 29341-6-17 |
| UPnP HVAC_ZoneThermostat:1 Device | ISO/IEC 29341-6-2 |

| | |
|---|---|
| UPnP BinaryLight:1 Device | ISO/IEC 29341-7-1 |
| UPnP Dimming:1 Service | ISO/IEC 29341-7-10 |
| UPnP SwitchPower:1 Service | ISO/IEC 29341-7-11 |
| UPnP DimmableLight:1 Device | ISO/IEC 29341-7-2 |
| UPnP InternetGatewayDevice:1 Device | ISO/IEC 29341-8-1 |
| UPnP LANHostConfigManagement:1 Service | ISO/IEC 29341-8-10 |
| UPnP Layer3Forwarding:1 Service | ISO/IEC 29341-8-11 |
| UPnP LinkAuthentication:1 Service | ISO/IEC 29341-8-12 |
| UPnP RadiusClient:1 Service | ISO/IEC 29341-8-13 |
| UPnP WANCableLinkConfig:1 Service | ISO/IEC 29341-8-14 |
| UPnP WANCommonInterfaceConfig:1 Service | ISO/IEC 29341-8-15 |
| UPnP WANDSLLinkConfig:1 Service | ISO/IEC 29341-8-16 |
| UPnP WANEthernetLinkConfig:1 Service | ISO/IEC 29341-8-17 |
| UPnP WANIPConnection:1 Service | ISO/IEC 29341-8-18 |
| UPnP WANPOTSLinkConfig:1 Service | ISO/IEC 29341-8-19 |
| UPnP LANDevice:1 Device | ISO/IEC 29341-8-2 |
| UPnP WANPPPConnection:1 Service | ISO/IEC 29341-8-20 |
| UPnP WLANConfiguration:1 Service | ISO/IEC 29341-8-21 |
| UPnP WANDevice:1 Device | ISO/IEC 29341-8-3 |
| UPnP WANConnectionDevice:1 Device | ISO/IEC 29341-8-4 |
| UPnP WLANAccessPointDevice:1 Device | ISO/IEC 29341-8-5 |
| UPnP Printer:1 Device | ISO/IEC 29341-9-1 |
| UPnP ExternalActivity:1 Service | ISO/IEC 29341-9-10 |
| UPnP Feeder:1.0 Service | ISO/IEC 29341-9-11 |
| UPnP PrintBasic:1 Service | ISO/IEC 29341-9-12 |
| UPnP Scan:1 Service | ISO/IEC 29341-9-13 |
| UPnP Scanner:1.0 Device | ISO/IEC 29341-9-2 |
| UPnP QoS Architecture:1.0 | ISO/IEC 29341-10-1 |
| UPnP QosDevice:1 Service | ISO/IEC 29341-10-10 |
| UPnP QosManager:1 Service | ISO/IEC 29341-10-11 |
| UPnP QosPolicyHolder:1 Service | ISO/IEC 29341-10-12 |
| UPnP QoS Architecture:2 | ISO/IEC 29341-11-1 |
| UPnP QosDevice:2 Service | ISO/IEC 29341-11-10 |
| UPnP QosManager:2 Service | ISO/IEC 29341-11-11 |
| UPnP QosPolicyHolder:2 Service | ISO/IEC 29341-11-12 |
| UPnP QOS v2 Schema Files | ISO/IEC 29341-11-2 |
| UPnP RemoteUIClientDevice:1 Device | ISO/IEC 29341-12-1 |
| UPnP RemoteUIClient:1 Service | ISO/IEC 29341-12-10 |
| UPnP RemoteUIServer:1 Service | ISO/IEC 29341-12-11 |
| UPnP RemoteUIServerDevice:1 Device | ISO/IEC 29341-12-2 |
| UPnP DeviceSecurity:1 Service | ISO/IEC 29341-13-10 |
| UPnP SecurityConsole:1 Service | ISO/IEC 29341-13-11 |
| UPnP ContentDirectory:3 Service | ISO/IEC 29341-14-12:2011 |
| UPnP MediaServer:3 Device | ISO/IEC 29341-14-3:2011 |
| UPnP ContentSync:1 | ISO/IEC 29341-15-10:2011 |
| UPnP Low Power Architecture:1 | ISO/IEC 29341-16-1:2011 |
| UPnP LowPowerProxy:1 Service | ISO/IEC 29341-16-10:2011 |

| | |
|---|---|
| UPnP LowPowerDevice:1 Service | ISO/IEC 29341-16-11:2011 |
| UPnP QoS Architecture:3 | ISO/IEC 29341-17-1:2011 |
| UPnP QosDevice:3 Service | ISO/IEC 29341-17-10:2011 |
| UPnP QosManager:3 Service | ISO/IEC 29341-17-11:2011 |
| UPnP QosPolicyHolder:3 Service | ISO/IEC 29341-17-12:2011 |
| UPnP QosDevice:3 Addendum | ISO/IEC 29341-17-13:2011 |
| UPnP RemoteAccessArchitecture:1 | ISO/IEC 29341-18-1:2011 |
| UPnP InboundConnectionConfig:1 Service | ISO/IEC 29341-18-10:2011 |
| UPnP RADAConfig:1 Service | ISO/IEC 29341-18-11:2011 |
| UPnP RADASync:1 Service | ISO/IEC 29341-18-12:2011 |
| UPnP RATAConfig:1 Service | ISO/IEC 29341-18-13:2011 |
| UPnP RAClient:1 Device | ISO/IEC 29341-18-2:2011 |
| UPnP RAServer:1 Device | ISO/IEC 29341-18-3:2011 |
| UPnP RADiscoveryAgent:1 Device | ISO/IEC 29341-18-4:2011 |
| UPnP SolarProtectionBlind:1 Device | ISO/IEC 29341-19-1:2011 |
| UPnP TwoWayMotionMotor:1 Service | ISO/IEC 29341-19-10:2011 |
| UPnP AV Architecture:2 | ISO/IEC 29341-20-1 |
| UPnP AVTransport:3 Service | ISO/IEC 29341-20-10 |
| UPnP ConnectionManager:3 Service | ISO/IEC 29341-20-11 |
| UPnP ContentDirectory:4 Device | ISO/IEC 29341-20-12 |
| UPnP RenderingControl:3 Service | ISO/IEC 29341-20-13 |
| UPnP ScheduledRecording:2 Service | ISO/IEC 29341-20-14 |
| UPnP MediaRenderer:3 Service | ISO/IEC 29341-20-2 |
| UPnP MediaServer:4 Device | ISO/IEC 29341-20-3 |
| UPnP AV Datastructure Template:1 | ISO/IEC 29341-20-4 |
| UPnP InternetGatewayDevice:2 Device | ISO/IEC 29341-24-1 |
| UPnP WANIPConnection:2 Service | ISO/IEC 29341-24-10 |
| UPnP WANIPv6FirewallControl:1 Service | ISO/IEC 29341-24-11 |
| UPnP WANConnectionDevice:2 Service | ISO/IEC 29341-24-2 |
| UPnP WANDevice:2 Device | ISO/IEC 29341-24-3 |
| UPnP Telephony Architecture:2 | ISO/IEC 29341-26-1 |
| UPnP CallManagement:2 Service | ISO/IEC 29341-26-10 |
| UPnP MediaManagement:2 Service | ISO/IEC 29341-26-11 |
| UPnP Messaging:2 Service | ISO/IEC 29341-26-12 |
| UPnP PhoneManagement:2 Service | ISO/IEC 29341-26-13 |
| UPnP AddressBook:1 Service | ISO/IEC 29341-26-14 |
| UPnP Calendar:1 Service | ISO/IEC 29341-26-15 |
| UPnP Presense:1 Service | ISO/IEC 29341-26-16 |
| UPnP TelephonyClient:2 Device | ISO/IEC 29341-26-2 |
| UPnP TelephonyServer:2 Device | ISO/IEC 29341-26-3 |
| UPnP Friendly Info Update:1 Service | ISO/IEC 29341-27-1 |
| UPnP MultiScreen MultiScreen Architecture:1 | ISO/IEC 29341-28-1 |
| UPnP MultiScreen Application Management:1 Service | ISO/IEC 29341-28-10 |
| UPnP MultiScreen Screen:1 Device | ISO/IEC 29341-28-2 |
| UPnP MultiScreen Application Management:2 Service | ISO/IEC 29341-29-10 |
| UPnP MultiScreen Screen:2 Device | ISO/IEC 29341-29-2 |
| UPnP IoT Management and Control Architecture Overview:1 | ISO/IEC 29341-30-1 |

| | |
|---|---|
| UPnP DataStore:1 Service | ISO/IEC 29341-30-10 |
| UPnP IoT Management and Control Data Model:1 Service | ISO/IEC 29341-30-11 |
| UPnP IoT Management and Control Transport Generic:1 Service | ISO/IEC 29341-30-12 |
| UPnP IoT Management and Control:1 Device | ISO/IEC 29341-30-2 |
| UPnP Energy Management:1 Service | ISO/IEC 29341-31-1 |

## 1 Scope

This part of Publicly Available Specification ISO/IEC 29341 specifies Sensor Data Models. It is intended to be an adjunct document to the IoT Management and Control Device [11] and DataStore Service [13] specifications.

## 2 Normative References

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

[1] UPnP Device Architecture, version 1.0, UPnP Forum, June 13, 2000. Available at: http://upnp.org/specs/arch/UPnPDA10_20000613.pdf. Latest version available at: http://upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.0.pdf.

[2] ISO 8601 Data elements and interchange formats – Information interchange -- Representation of dates and times, International Standards Organization, December 21, 2000. Available at: ISO 8601:2000.

[3] IETF RFC 2119, Key words for use in RFCs to Indicate Requirement Levels, S. Bradner, 1997. Available at: http://www.faqs.org/rfcs/rfc2119.html.

[4] HyperText Transport Protocol – HTTP/1.1, R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, June 1999. Available at: http://www.ietf.org/rfc/rfc2616.txt.

[5] IETF RFC 3339, Date and Time on the Internet: Timestamps, G. Klyne, Clearswift Corporation, C. Newman, Sun Microsystems, July 2002. Available at: http://www.ietf.org/rfc/rfc3339.txt.

[6] Extensible Markup Language (XML) 1.0 (Third Edition), François Yergeau, Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, eds., W3C Recommendation, February 4, 2004. Available at: http://www.w3.org/TR/2004/REC-xml-20040204.

[7] XML Schema Part 2: Data Types, Second Edition, Paul V. Biron, Ashok Malhotra, W3C Recommendation, 28 October 2004. Available at: http://www.w3.org/TR/2004/REC-xmlschema-2-20041028.

[8] ISO/IEEE-11073-20601 Health informatics - Personal health device communication - Part 20601: Application Profile - Optimized exchange protocol, 2010. Available at: http://www.iso.org/iso/search.htm?qt=11073&searchSubmit=Search&sort=rel&type=simple&published=true

[9] ISO/IEEE-11073-104zz Health informatics -- Personal health device communication -- Part 104zz: Device specialization -- <see specific specialization>. Available at: http://www.iso.org/iso/search.htm?qt=11073&searchSubmit=Search&sort=rel&type=simple&published=true

[10] UPnP IoT Management and Control Architecture Overview, UPnP Forum, July 1, 2013. Available at: http://www.upnp.org/specs/iotmc/UPnP-iotmc-IoTManagementAndControl-Architecture-Overview-v1-20130701.pdf. Latest version available at: http://www.upnp.org/specs/iotmc/UPnP-iotmc-IoTManagementAndControl-Architecture-Overview-v1.pdf.

[11] UPnP IoT Management and Control Device, UPnP Forum July 1, 2013. Available at: http://www.upnp.org/specs/iotmc/UPnP-iotmc-IoTManagementAndControl-v1-Device-20130701.pdf. Latest version available at: http://www.upnp.org/specs/iotmc/UPnP-iotmc-IoTManagementAndControl-v1-Device.pdf.

[12] UPnP IoT Management and Control Transport Generic:1 Service, UPnP Forum July 1, 2013. Available at: http://www.upnp.org/specs/iotmc/UPnP-iotmc-IoTManagementAndControl-TransportGeneric-v1-Service-20130701.pdf. Latest version available

at: http://www.upnp.org/specs/iotmc/UPnP-iotmc-IoTManagementAndControl-TransportGeneric-v1-Service.pdf.

[13] UPnP DataStore:1 Service, UPnP Forum, July 1, 2013. Available at: http://www.upnp.org/specs/smgt/UPnP-ds-DataStore-v1-Service-20130701.pdf. Latest version available at: http://www.upnp.org/specs/ds/UPnP-smgt-DataStore-v1-Service.pdf.

[14] UPnP IoT Management and Control Sensor DataModel Service, UPnP Forum, July 1, 2013. Available at: http://www.upnp.org/specs/iotmc/UPnP-iotmc-IoTManagementAndControl-DataModel-v1-Service-20130701.pdf. Latest version available at: http://www.upnp.org/specs/iotmc/UPnP-iotmc-IoTManagementAndControl-DataModel-v1-Service.pdf.

[15] UPnP DeviceProtection:1 Service, UPnP Forum, February 24, 2011. Available at: http://www.upnp.org/specs/gw/UPnP-gw-DeviceProtection-v1-Service-20110224.pdf.
Latest version available at: http://www.upnp.org/specs/gw/UPnP-gw-DeviceProtection-v1-Service.pdf.

[16] UPnP ConfigurationManagement:2 Service, UPnP Forum, December 31, 2010. Available at: http://www.upnp.org/specs/dm/UPnP-dm-ConfigurationManagement-v2-Service-20120216.pdf. Latest version available at: http://www.upnp.org/specs/dm/UPnP-dm-ConfigurationManagement-v2-Service.pdf.

[17] XML Schema UPnP IoT Management and Control Sensor DataModel Events, UPnP Forum, July 1, 2013. Available at: http://www.upnp.org/schemas/smgt/sdmevent-v1-20130701.xsd. Latest version available at: http://www.upnp.org/schemas/smgt/sdmevent.xsd.

[18] XML Schema UPnP IoT Management and Control Sensor DataModel DataItem Description, UPnP Forum, July 1, 2013. Available at: http://www.upnp.org/schemas/smgt/sdmdid-v1-20130701.xsd. Latest version available at: http://www.upnp.org/schemas/smgt/sdmdid.xsd.

[19] XML Schema UPnP IoT Management and Control DataRecord Information, UPnP Forum, July 1, 2013. Available at: http://www.upnp.org/schemas/smgt/srecinfo-v1-20130701.xsd. Latest version available at: http://www.upnp.org/schemas/smgt/srecinfo.xsd.

[20] ISO/IEC 14977, *Information technology - Syntactic metalanguage - Extended BNF*, December 1996.

## 3 Terms, Definitions and Abbreviations

For the purposes of this document, the terms and definitions given in UPnP Device Architecture, version 1.0 [1] apply.

## 4 Notations and conventions

### 4.1 Notation

- Strings that are to be taken literally are enclosed in “double quotes”.
- Words that are emphasized are printed in *italic*.
- Keywords that are defined by the UPnP Working Committee are printed using the *forum* character style.
- Keywords that are defined by the UPnP Device Architecture are printed using the **arch** character style.
- A double colon delimiter, “::”, signifies a hierarchical parent-child (parent::child) relationship between the two objects separated by the double colon. This delimiter is used in multiple contexts, for example: Service::Action(), Action()::Argument, parentProperty::childProperty.

## 4.2 Data Types

This specification uses data type definitions from two different sources. The UPnP Device Architecture defined data types are used to define state variable and action argument data types [1]. The XML Schema namespace is used to define property data types [7].

For UPnP Device Architecture defined Boolean data types, it is strongly RECOMMENDED to use the value “**0**” for false, and the value “**1**” for true. The values “**true**”, “**yes**”, “**false**”, or “**no**” MAY also be used but are NOT RECOMMENDED. The values “**yes**” and “**no**” are deprecated and MUST NOT be sent out by devices but MUST be accepted on input.

For XML Schema defined Boolean data types, it is strongly RECOMMENDED to use the value “*0*” for false, and the value “*1*” for true. The values “*true*”, “*yes*”, “*false*”, or “*no*” MAY also be used but are NOT RECOMMENDED. The values “*yes*” and “*no*” are deprecated and MUST NOT be sent out by devices but MUST be accepted on input.

## 4.3 Vendor-defined Extensions

Whenever vendors create additional vendor-defined state variables, actions or properties, their assigned names and XML representation MUST follow the naming conventions and XML rules as specified in [1], Clause 2.5, “Description: Non-standard vendor extensions”.

### 4.3.1 Extended Backus-Naur Form

Extended Backus-Naur Form is used in this document for a formal syntax description of certain constructs. The usage here is according to the reference [20].

#### 4.3.1.1 Typographic conventions for EBNF

`Non-terminal` symbols are unquoted sequences of characters from the set of English upper and lower case letters, the digits “`0`” through “`9`”, and the hyphen (“`-`”). Character sequences between `'single quotes'` are terminal strings and shall appear literally in valid strings. Character sequences between `(*comment delimiters*)` are English language definitions or supplementary explanations of their associated symbols. White space in the EBNF is used to separate elements of the EBNF, not to represent white space in valid strings. White space usage in valid strings is described explicitly in the EBNF. Finally, the EBNF uses the following operators:

**Table 4-1 — EBNF Operators**

| Operator | Semantics |
|---|---|
| ::= | **definition** – the non-terminal symbol on the left is defined by one or more alternative sequences of terminals and/or non-terminals to its right. |
| \| | **alternative separator** – separates sequences on the right that are independently allowed definitions for the non-terminal on the left. |
| * | **null repetition** – means the expression to its left may occur zero or more times. |
| + | **non-null repetition** – means the expression to its left shall occur at least once and may occur more times. |
| [ ] | **optional –** the expression between the brackets is allowed. |
| ( ) | **grouping –** groups the expressions between the parentheses. |
| - | **character range** – represents all characters between the left and right character operands inclusively. |

# 5 IoT Management and Control Configuration Management Service Profile

## 5.1 Service Type

This specification reuses the *ConfigurationManagement* service exactly as specified in ConfigurationManagement service [16]:

**urn:schemas-upnp-org:service:ConfigurationManagement:2**

*IoT Management and Control* DataModel service defined in this specification refers to the same service type.

## 5.2 IoT Management and Control Device Architecture

The UPnP IoT Management and Control device [11] provides UPnP client access to Sensors and Actuators. These devices may be connected directly to the UPnP network or can be connected to non-UPnP based networks which are bridged by this device into the UPnP network. The IoT Management and Control device includes a UPnP ConfigurationManagement service [16] which describes Sensors to UPnP based clients.

### 5.2.1 Sensor Description

The UPnP IoT Management and Control DataModel service provides a set of uniform Sensor Properties as defined by Annex A, "IoT Management and Control General Data Model". These properties assist UPnP clients to identify sensors they may be capable of supporting. In addition to uniform Sensor properties described by the General Sensor Data Model, this specification also can reference additional sensor properties which are defined by the Sensor's parent ecosystem.

### 5.2.2 Sensor Data Interface

The UPnP IoT Management and Control device [11] also includes a UPnP IoT Management and Control TransportGeneric service [12]. This service treats Sensors and Actuators as abstract data sources and sinks. While the IoT Management and Control device performs much of the interfacing tasks for Sensors, the actual data formats delivered by the Sensors are largely determined by the parent ecosystems for each target Sensor or Actuator. UPnP describes Sensor and Actuator data inputs/outputs as named DataItems(s). Sensor DataItem(s) are described by providing a name, format, encoding (see UPnP IoT Management and Control Architecture Overview [10], subclause 4.3, "DataItem Semantics" for additional information).

### 5.2.3 Sensor Data Records

A UPnP client may transfer data to/from a Sensor(s) supported by the IoT Management and Control device using actions in the included UPnP IoT Management and Control TransportGeneric service [12]. When requesting data, the UPnP client specifies an XML document conforming to the XML Schema UPnP DataRecord Information [19]. This document identifies a series of one or more DataItem(s) supported by the target Sensor. The UPnP client may either read or write the Sensor directly via SOAP actions or may establish a transport connection allowing the Sensor to deliver DataRecord(s) directly to a transport client.

### 5.2.4 SensorURNs

A SensorURN identifies a set of DataItem(s) a Sensor supports. See subclause 5.2.6, "Sensor Normative Type Identifiers" for requirements on construction of SensorURN values.

### 5.2.5 Sensor Required DataItems

Implementation of certain DataItem(s) is required by the IoT Management and Control specification. These DataItem(s) shall be available for all Sensor URNs whether or not explicitly defined by the corresponding SensorURN definition. The required DataItem(s) are listed in Annex B "Required Sensor DataItems".

### 5.2.6 Sensor Normative Type Identifiers

UPnP IoT Management and Control defines a uniform template for generating normative identifiers. These identifiers include SensorCollectionType, SensorType and SensorURNs, The following template is used:

SensorIdentifier ::= "urn:" upnp-org ":" [identifier-type] ":" [generic-device-type] ":" [sensor-vendor-identifier] ":" [sensor-vendor-spec] ":" [vendor-identifier] : [vendor-model-identifier] ":" [identifier-type-dependent]

SensorIdentifier(s) using the [upnp-org] domain shall use the following [identifier-type] values:

**Table 5-1 — Sensor URN [identifier-type] values**

| Sensor Identifier | [identifier-type] value |
|---|---|
| Sensor Collection Type | *smgt-sct* |
| Sensor Type | *smgt-st* |
| Sensor URN | *smgt-surn* |

The [generic-device-identifiers] field is descriptive. UPnP IoT Management and Control defines device-identifiers for various common devices (see Annex C, "Common Device Identifiers"). If one of these values is suitable, then the UPnP defined value should be used. However, if a sensor application is unique, vendor-specific extensions are permitted.

The [sensor-vendor-identifier] shall contain the ICANN domain-name of the organization or vendor which defines the sensor specification. This name shall identify the vendor which defines the sensors for the device rather than the final integrator of the device.

The [sensor-vendor-spec] shall identify the specification the sensor conforms to.

The [vendor-identifier] shall contain the ICAN domain-name of the device vendor. This domain-name shall identify the vendor which the customer would normally identify as the manufacturer of the device.

The [vendor-model-identifier] shall provide vendor specific identifying information for the device.

The [identifier-type-dependent] field is defined based on the type of SensorIdentifier as follows:

**Table 5-2 — Sensor URN [identifier-type-dependent] values**

| [identifier-type-dependent] | [identifier-types] | Description |
|---|---|---|
| *monitor* | *smgt-st, smgt-surn* | DataItem(s) or Sensor(s) intended to periodically monitor or record device performance over a period of time. |
| *status* | *smgt-st, smgt-surn* | DataItem(s) or Sensor(s) intended to provide current device status or settings. |
| *datafeed* | *smgt-st, smgt-surn* | DataItem(s) or Sensor(s) which to provide a continuous source of sensor data. |
| *telemetry* | *smgt-st, smgt-surn* | DataItem(s) or Sensor(s) intended to report data obtained by a Sensor. |
| *setting* | *smgt-st, smgt-surn* | DataItem(s) or Sensor(s) intended to modify end-user settable characteristics of a device. |
| *control* | *smgt-st, smgt-surn* | DataItem(s) or Sensor(s) intended to control aspects of a device not typically set by end-users |
| *alarms* | *smgt-st, smgt-surn* | DataItem(s) or Sensor(s) intended to convey abnormal conditions for a device. |

### 5.2.7 Event Model

The IoT Management and Control event model generates two categories of events.

- Sensor Configuration events

  Configuration events indicate addition or removal of SensorCollection(s) or Sensor(s) within a collection. Reporting of Sensor and SensorCollection configuration events are always enabled.

- Sensor Specific events

  Specific events indicate state changes to Sensor(s) such as data availability or transport connection errors. Reporting of Sensor events may be enabled on a per-Sensor basis via the *SensorEventsEnable* parameter.

Sensor events are reported via the *SensorEvents* parameter which provides an XML document conforming to XML Schema UPnP IoT Management and Control DataModel Events [17]. This parameter shall support the *EventOnChange* attribute value which shall be enabled by default and the *AlarmOnChange* attribute value which shall also be enabled by default.

Changes to the *SensorEvents* parameter shall be sent to subscribed control points via the ConfigurationManagement service *ConfigurationUpdate* state variable. The IOT Management and Control ConfigurationManagement service shall implement and set the *AlarmsEnabled* state variable to ("*1*") by default.

To maintain privacy and security sensor collection(s) and sensor(s) are only identified by their *CollectionID* and *SensorID* values. In order to obtain additional information a UPnP control point would need sufficient privileges to read to the corresponding Data Model nodes which describe the sensor collection or sensor.

The SensorEvents parameter value shall be set to an XML document conforming to the UPnP IoT Management and Control DataModel Events schema [17] and as described below:

```
<?xml version="1.0" encoding="utf-8"?>
<SensorEvents
   xmlns="urn:schemas-upnp-org:smgt:sdmevent"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:schemas-upnp-org:smgt:sdmevent
   http://www.upnp.org/schemas/smgt/sdmevent-v1.xsd">
   <sensorevent
      collectionID="sensorCollectionID"
      sensorID=”sensorID”
      event="sensor event or sensor collection event name "/>
    ...
</SensorEvents>
```

`<?xml>`

Required. Case Sensitive

`<SensorEvents>`

Required. Shall include the namespace declaration for the SensorEvents XML Schema (“urn:schemas-upnp-org:smgt:sdmevent”). Shall include the zero or more of the following element and attributes:

`<sensorevent>`

Required. Shall occur zero or more times. Indicates that a sensor or sensor collection event has occurred.

`collectionID`

Required. Indicates the *CollectionID* for the reported event.

`sensorID`

Required. Indicates the *SensorID* for the reported event. If there is no corresponding *SensorID* for a reported event this attribute shall be set to "".

`event`

Required. Indicates the type of sensor related event. The `event` attribute shall use one of the values as detailed in the following table.

### Table 5-3 — <SensorEvents> `event=` attribute allowed values

| SensorEvent Name | Description |
|---|---|
| *SensorCollectionAdded* | A new sensor collection has been added.<br>Reporting of this event is enabled by default. |
| *SensorCollectionRemoved* | A previously reported sensor collection has been removed. Reporting of this event is enabled by default. |
| *SensorAdded* | A new sensor has been added.<br>Reporting of this event is enabled by default. |
| *SensorRemoved* | A previously reported sensor has been removed.<br>Reporting of this event is enabled by default. |
| *SOAPDataAvailable* | The sensor has data available to be read by the *ReadSensor()* SOAP action.<br>Reporting of this event is disabled by default. Reporting of this event can be enabled by setting the *SensorEventsControl* parameter to "SOAPDataAvailableEnable,1". |
| *SOAPDataOverrun* | The sensor has discarded data which was not read by any SOAP client.<br>Reporting of this event is disabled by default. Reporting of this event can be enabled by setting the *SensorEventControl* parameter to "SOAPDataOverrunEnable,1". |
| *TransportDataAvailable* | The sensor has received data available to be sent to one or more transport connection for this senor.<br>Reporting of this event is disabled by default. Reporting of this event can be enabled by setting the *SensorEventControl* parameter to "TransportDataAvailableEnable,1". |
| *TransportDataOverrun* | The sensor has discarded data which was not read by any active transport connection.<br>Reporting of this event is disabled by default. Reporting of this event can be enabled by setting the *SensorEventControl* parameter to "TransportDataOverrunEnable,1". |
| *TransportConnectionError* | An error has occurred on one or more active transport connections for this sensor.<br>Reporting of this event is disabled by default. Reporting of this event can be enabled by setting the *SensorEventControl* parameter to "TransportConnectionErrorEnable,1". |
| *SensorAvailable* | A currently configured sensor has become available.<br>Reporting of this event is disabled by default. Reporting of this event can be enabled by setting the *SensorEventControl* parameter to "SensorAvailabilityEnable,1". |
| *SensorUnavailable* | A currently configured sensor has become unavailable, Reporting of this event is disabled by default. Reporting of this event can be enabled by setting the *SensorEventControl* parameter to "SensorAvailabilityEnable,1". |

## 5.3 State Variables

The IoT Management and Control device [11] reuses some of the state variables defined in the ConfigurationManagement service [16].

The Table 5-4 below lists all the eventable state variables used in the *IoT Management and Control* device. These state variables are defined in the CMS. The table also indicates the required/allowed constraint for the *IoT Management and Control* device.

All the required state variables from the CMS shall be implemented by *IoT Management and Control*, as specified in [1]. It is up to the implementation to choose whether to support allowed state variables.

The *AttributeValuesUpdate* state variable should be supported if the CMS implementation support the *SetAttributeValues()* for changing attribute values (i.e. the values of *EventOnChange* and *Version*, when it is also supported).

The *IoT Management and Control* device supports the A_ARG_TYPE state variables defined in the CMS depending on the required/allowed constraint of the respective actions for the *IoT Management and Control*. The A_ARG_TYPE_ state variables are not listed in this document, refer to the ConfigurationManagement service [16] for the details.

**Table 5-4 — State Variables for Eventing**

| Variable Name | CMS<br>R/A [a] | IOTManagement and Control<br>R/A [a] |
|---|---|---|
| *ConfigurationUpdate* | *R* | *R* |
| *CurrentConfigurationVersion* | *R* | *R* |
| *SupportedDataModelsUpdate* | *R* | *R* |
| *SupportedParametersUpdate* | *R* | *R* |
| *AttributeValuesUpdate* | *A* | *A* |
| *InconsistentStatus* | *A* | *A* |
| *AlarmsEnabled* | *CR* | *R* |

[a] *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*).

Note: For first-time reader, it may be more insightful to read the theory of operations first and then the action definitions before reading the state variable definitions.

## 5.4 Actions

### 5.4.1 Introduction

The IoT Management and Control device reuses the actions defined in the CMS. Some of the actions which are allowed in CMS are required or conditionally required for *IoT Management and Control*.

Table 5-5 below lists all the CMS actions, with the following additional information:

- The column “CMS R/A” indicates whether the action is required or allowed in ConfigurationManagement service [16]; all the required actions from the CMS shall be implemented by the device supporting IOT Management and Control.
- The column “IoT Management and Control R/A” indicates whether the action is either required or allowed for the IoT Management and Control DataModel service [14]; some of the allowed actions are required for IoT Management and Control.
- The column “IoT Management and Control CP R/A” indicates whether the IoT Management and Control CP shall support the action to be fully compliant with the IoT Management and Control device specifications.

Refer to ConfigurationManagement service [16] for more detailed information on each action. The remaining subclauses of 5.4 give additional information on some of the actions as used by the IoT Management and Control profile.

**Table 5-5 — Actions**

| Name | CMS<br>R/A [a] | IoT Management and Control<br>R/A [b] | IoT Management and Control CP<br>R/A [c] |
|---|---|---|---|
| *GetSupportedDataModels()* | *R* | *R* | *R* |
| *GetSupportedParameters()* | *R* | *R* | *R* |
| *GetInstances()* | *R* | *R* | *R* |
| *GetValues()* | *R* | *R* | *R* |
| *GetSelectedValues()* | *A* | *A* | *A* |
| *SetValues()* | *A* | *R* | *A* |
| *CreateInstance()* | *A* | *CR* | *A* |
| *DeleteInstance()* | *A* | *CR* | *A* |
| *GetAttributes()* | *R* | *R* | *A* |
| *SetAttributes()* | *A* | *A* | *A* |
| *GetInconsistentStatus()* | *A* | *A* | *A* |
| *GetConfigurationUpdate()* | *R* | *R* | *A* |
| *GetCurrentConfigurationVersion()* | *R* | *R* | *A* |
| *GetSupportedDataModelsUpdate()* | *R* | *R* | *A* |
| *GetSupportedParametersUpdate()* | *R* | *R* | *A* |
| *GetAttributeValuesUpdate()* | *A* | *A* | *A* |

[a] For a device this column indicates whether the action shall be implemented or not, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*).

[b] For a device this column indicates whether the action shall be implemented or not, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*).

[c] For a control point this column indicates whether a control point shall be capable of invoking this action, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*).

### 5.4.2 *GetSupportedDataModels()*

The IoT Management and Control implementation shall provide an XML document compliant with the `<SupportedDataModels>` root element per the CMS namespace (urn:schemas-upnp-org:dm:cms). This element shall contain a `<SubTree>` child element which describing the IoT Management and Control DataModel as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<cms:SupportedDataModels
   xmlns:cms="urn:schemas-upnp-org:dm:cms"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn: schemas-upnp-org:dm:cms
                        http://www.upnp.org/schemas/dm/cms.xsd">
   <SubTree>
      <URI>
         urn:upnp-org:smgt:1
      </URI>
      <Location>
         /UPnP/SensorMgt
      </Location>
      <URL>
      http://www.upnp.org/specs/smgt/UPnP-smgt-SensorDataModel-v1-Service.pdf
      </URL>
      <Description>
         ... device vendor descriptive text ...
      </Description>
   </SubTree>
</SupportedDataModels>
```

`<?xml>`

Required. Case Sensitive

`<SupportedDataModels>`

Required. Shall include the name space declaration for the CMS XML Schema ("urn:schemas-upnp-org:dm:cms").

`<SubTree>`

Required. Shall include a `<SubTree>` element with the following elements:

`<URI>`

Required. Implementations complying with this specification shall provide the value:

`urn:upnp-org:smgt:1`

`<Location>`

Required. Implementations complying with this specification shall provide the value:

`/UPnP/SensorMgt`

`<URL>`

Allowed. Implementations may use the following value default value corresponding to the <URI> value (above) or may provide a reference to device vendor supplied specifications:

`http://www.upnp.org/specs/smgt/UPnP-smgt-SensorDataModel-v1-Service.pdf`

`<Description>`

Allowed. Implementations may provide descriptive text for their IoT Management and Control implementation or may omit this element.

### 5.4.3 *GetSelectedValues()*

The *GetSelectedValues()* action is an allowed action for the CMS, and also an allowed action in IoT Management and Control .

### 5.4.4 *SetValues()*

The *SetValues()* action is an allowed action for the CMS, but from the IoT Management and Control perspective it is a required action and shall be implemented.

This action is required since the IoT Management and Control Device DataModel service supports Parameters with write access. This action allows a IoT Management and Control CP to change the values of such Parameters.

### 5.4.5 *CreateInstance()*

The *CreateInstance()* action is an allowed action for the CMS, but the IoT Management and Control Device perspective it is a conditionally required action. This action shall be implemented if the IoT Management and Control device supports control point additions and/or deletions to a sensor's SensorGroups and/or SensorPermissions MultiInstance parameter nodes.

### 5.4.6 *DeleteInstance()*

The *DeleteInstance()* action is an allowed action for the CMS, but the Iot Management and Control perspective it is a conditionally required action. This action shall be implemented if the IoT Management and Control device supports control point additions and/or deletions to a Sensor's SensorGroups and/or SensorPermissions MultiInstance parameter nodes.

### 5.4.7 *SetAttributes()*

The *SetAttributes()* action is an allowed action for the CMS, and is also an allowed action for IoT Management and Control.

# Annex A IoT Management and Control General Data Model (normative)

### Table A.1 — IoT Management and Control General Parameters

| Name | Type | Acc | Req | Reference | EOC | Ver |
|---|---|---|---|---|---|---|
| /UPnP/SensorMgt/ | SingleInstance | RO | R | A.1.1.1 | | 0 |
| SensorEvents | string | RO | R | A.1.1.2 | 1 | 1 |
| SensorCollectionsNumberOfEntries | int | RO | R | A.1.1.3 | | 0 |
| | | | | | | |
| /UPnP/SensorMgt/SensorCollections/#/ | MultiInstance | RO | R | A.1.1.4 | | 0 |
| CollectionID | string | RO | R | A.1.1.5 | | 0 |
| CollectionType | string | RO | R | A.1.1.6 | | 0 |
| CollectionFriendlyName | string | RW | R | A.1.1.7 | | 0 |
| CollectionInformation | string | RW | R | A.1.1.8 | | 0 |
| CollectionUniqueIdentifier | string | RO | R | A.1.1.9 | | 0 |
| CollectionSpecific/ | SingleInstance | RO | CR | A.1.1.10 | | 0 |
| SensorsNumberOfEntries | int | RO | R | A.1.1.11 | | 0 |
| | | | | | | |
| /UPnP/SensorMgt/SensorCollections/#/Sensors/#/ | MultiInstance | RO | R | A.1.1.12 | | 0 |
| SensorID | string | RO | R | A.1.1.13 | | 0 |
| SensorType | string | RO | R | A.1.1.14 | | 0 |
| SensorUniqueIdentifier | string | RO | A | A.1.1.15 | | 0 |
| SensorUpdateRequest | boolean | RW | A | A.1.1.15 | | 0 |
| SensorPollingInterval | Int | RW | A | A.1.1.16 | | 0 |
| SensorReportChangeOnly | boolean | RW | A | A.1.1.17 | | 0 |
| SensorEventsEnable | string | RW | A | A.1.1.18 | | 0 |
| SensorSpecific/ | SingleInstance | RO | CR | A.1.1.29 | | 0 |
| SensorRelatedNumberOfEntries | Int | RO | CA | A.1.1.19 | | 0 |
| SensorGroupsNumberOfEntries | Int | RO | CA | A.1.1.20 | | 0 |
| SensorDefaultPermissionsNumberOfEntries | Int | RO | CA | A.1.1.21 | | 0 |
| SensorURNsNumberOfEntries | Int | RO | R | A.1.1.30 | | 0 |
| | | | | | | |
| UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorRelated/#/ | MultiInstance | RO | A | A.1.1.22 | | 0 |
| SensorPath | string | RO | R | A.1.1.23 | | 0 |
| | | | | | | |
| UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorGroups/#/ | MultiInstance | RO | CR | A.1.1.24 | | 0 |
| SensorGroup | string | RW | R | A.1.1.25 | | 0 |
| | | | | | | |
| /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorDefaultPermissions/#/ | MultiInstance | RW | CA | A.1.1.26 | | 0 |
| SensorDefaultRole | string | RO | R | A.1.1.27 | | 0 |
| SensorDefaultPermissions | CSV(string) | RO | R | A.1.1.28 | | 0 |
| | | | | | | |
| /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorURNs/#/ | MultiInstance | RO | R | A.1.1.31 | | 0 |

| | | | | | | |
|---|---|---|---|---|---|---|
| SensorURN | string | RO | R | A.1.1.32 | | 0 |
| DataItemsNumberOfEntries | Int | RO | R | A.1.1.33 | | 0 |
| | | | | | | |
| /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorURNs/#/DataItems/#/ | MultiInstance | RO | R | A.1.1.34 | | 0 |
| Name | string | RO | R | A.1.1.35 | | 0 |
| Type | string | RO | R | A.1.1.36 | | 0 |
| Encoding | string | RO | R | A.1.1.37 | | 0 |
| Description | string | RO | R | A.1.1.38 | | 0 |
| | | | | | | |
| /UPnP/SensorMgt/SensorGroupSets/#/ | MultiInstance | RW | CR | A.1.1.39 | | 0 |
| GroupSetID | string | RW | R | A.1.1.40 | | 0 |
| GroupSetType | string | RW | R | A.1.1.41 | | 0 |
| GroupSetFriendName | string | RW | R | A.1.1.42 | | 0 |
| GroupSetSpecific/ | SingleInstance | RW | CR | A.1.1.43 | | 0 |
| MembersNumberOfEntries | int | RW | R | A.1.1.44 | | 0 |
| | | | | | | |
| /UPnP/SensorMgt/SensorGroupSets/#/Members/#/ | MultiInstance | RO | CR | A.1.1.45 | | 0 |
| MemberID (CollectionID or SensorID) | string | RW | R | A.1.1.46 | | 0 |
| | | | | | | |
| /UPnP/SensorMgt/SensorCollections/#/ | MultiInstance | RO | R | A.1.1.4 | | 0 |
| ParticipatingGroupSetID | string | RW | CR | A.1.1.47 | | 0 |
| | | | | | | |
| /UPnP/SensorMgt/SensorCollections/#/Sensors/#/ | MultiInstance | RO | R | A.1.1.12A.1.1.12 | | 0 |
| ParticipatingGroupSetID | String | RW | CR | A.1.1.47 | | 0 |
| | | | | | | |

## A.1 Introduction

This section describes IoT Management and Control Nodes and Parameters which are common to all sensor collections and sensors supported by the host IoT Management and Control device. Individual sensor collections appear as instances of the */UPnP/SensorMgt/SensorCollections/* MultiInstance node. Nodes which are sensor collection type specific are listed under the */UPnP/SensorMgt/SensorCollections/#/CollectionSpecific/* node. Each *SensorCollection* node hosts zero or more *Sensors nodes* which appear as instances of the */UPnP/SensorMgt/SensorCollections/#/Sensors/ MultiInstance* node. Nodes which are sensor type specific are listed under the */UPnP/SensorMgt/SensorCollections/#/SensorSpecific/* node for each *Sensor* instance. Each Sensor supports zero or more Sensor URNs */UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorURNs/#/SensorURNs* which identifies sets required and permitted DataItem(s) generated by each *Sensor*.

### A.1.1 IoT Management and Control General Parameters

#### A.1.1.1 /UPnP/SensorMgt/

**Type:** SingleInstance
**Description:** The required *SensorMgt* SingleInstance node identifies the root node of the IoT Management and Control DataModel.

#### A.1.1.2 /UPnP/SensorMgt/SensorEvents

**Type:** string

**Description:** The required *SensorEvents* parameter shall provide an XML document conforming to the XML Schema UPnP IoT Management and Control DataModel Events [17]. This document shall contain zero or more `<sensorEvent>` elements where each element occurrence reports a unique event condition as indicated by the `event` attribute for the *SensorCollectionID* and *SensorID* identified by the `collectionID` and `sensorID` attributes. This parameter shall remain unchanged for a minimum event moderation period (200mS). If additional events have accumulated a new SensorEvents XML document shall replace the current document after any active event moderation period has expired.

#### A.1.1.3 /UPnP/SensorMgt/SensorCollectionsNumberOfEntries

**Type:** int

**Description:** The required *SensorCollectionsNumberOfEntries parameter* shall provide the number of *SensorCollections* MultiInstance node entries.

#### A.1.1.4 /UPnP/SensorMgt/SensorCollections/

**Type:** MultiInstance

**Description:** The required *SensorCollections* MultiInstance node identifies zero or more collections containing sensors and/or actuators supported by this IoT Management and Control device.

#### A.1.1.5 /UPnP/SensorMgt/SensorCollections/#/CollectionID

**Type:** string

**Description:** The required *CollectionID parameter* provides a unique identifier for the purpose of issuing SOAP actions or identifying the sensor collection associated with events reported by the TransportGeneric service [12].

#### A.1.1.6 /UPnP/SensorMgt/SensorCollections/#/CollectionType

**Type:** string

**Description:** The required *CollectionType* parameter value describes the type of sensor for this *SensorCollections* node instance. See the subclause 5.2.5, "Sensor Normative Type Identifiers" for required and allowed values for this parameter.

#### A.1.1.7 /UPnP/SensorMgt/SensorCollections/#/CollectionFriendlyName

**Type:** string

**Description:** The required *CollectionFriendlyName* parameter provides a string identifying the sensor collection suitable for end-user display. The IoT Management and Control service is allowed to preset this parameter. The IoT Management and Control service shall allow this field to be written provided the requesting control point has sufficient privileges. The IoT Management and Control service should retain previously written values between reboots.

#### A.1.1.8 /UPnP/SensorMgt/SensorCollections/#/CollectionInformation

**Type:** string

**Description:** The required *CollectionInformation* parameter provides a string describing the sensor collection suitable for end-user display. The IoT Management and Control service is allowed to preset this parameter. The IoT Management and Control service shall permit this parameter to be written provided the requesting control point has sufficient privileges. The IoT Management and Control service should retain previously written values between reboots.

#### A.1.1.9 /UPnP/SensorMgt/SensorCollections/#/CollectionUniqueIdentifier

**Type:** string

**Description:** The required *CollectionUniqueIdentifier* parameter provides a string which uniquely identifies a sensor collection. For devices discoverable on a UPnP network, this value shall match the UDN of the corresponding UPnP Device. See sections describing specific classes of sensor collection(s) for other allowed values.

### A.1.1.10 /UPnP/SensorMgt/SensorCollections/#/CollectionSpecific/

**Type:** SingleInstance

**Description:** The conditionally required *CollectionSpecific* SingleInstance node provides ecosystem specific sensor collection information. The child nodes of the *CollectionSpecific* node depend on the *CollectionType* parameter value. See the sections describing specific classes of sensor collection(s) for required and allowed nodes for specific sensor collection types.

### A.1.1.11 /UPnP/SensorMgt/SensorCollections/SensorsNumberOfEntries

**Type:** int

**Description:** The required *SensorsNumberOfEntrie*s parameter shall provide the number of *Sensor* MultiInstance node entries.

### A.1.1.12 /UPnP/SensorMgt/SensorCollections/#/Sensors/

**Type:** MultiInstance

**Description:** The required *Sensors* MultiInstance node shall describe a zero or more sensor(s) or actuator(s) managed by this service.

### A.1.1.13 /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorID

**Type:** string

**Description:** The required *SensorID* parameter provides a unique identifier for the purpose of issuing SOAP actions to the sensor. See the *A_ARG_TYPE_SensorID* in the TransportGeneric service specification [12] for further usage information about this parameter. The IoT Management and Control device is permitted to change assigned *SensorID* values between service restarts.

### A.1.1.14 /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorType

**Type:** string

**Description:** The required *SensorType* parameter value describes the type of sensor for this *Sensors* node instance. See the subclause 5.2.5, "Sensor Normative Type Identifiers" for required and allowed values for this parameter.

### A.1.1.15 /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorUpdateRequest

**Type:** boolean

**Description:** The allowed *SensorUpdateRequest* parameter when written to (“1”) shall cause the IoT Management and Control DataModel service to request an update from the sensor associated with this Sensors instance. When the IoT Management and Control DataModel service completes the update process, it shall reset this parameter to (“0”). See the sections describing specific classes of sensor collection(s) to determine if implementation of this parameter is required.

### A.1.1.16 /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorPollingInterval

**Type:** int

**Description:** The allowed *SensorPollingInterval* parameter when written to a non-zero value (milliseconds) shall cause IoT Management and Control service to request periodic updates from the sensor associated with this *Sensors* instance.

### A.1.1.17 /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorReportChangeOnly

**Type:** int

**Description:** This allowed *SensorReportChangeOnly* parameter when written to a (“1”) in addition to a non-zero *SensorPollingInterval* parameter shall cause the IoT Management and Control service to suppress reporting of identical sensor readings. The setting of this property shall not suppress sensor updates requested by setting the *SensorUpdateRequest* parameter.

### A.1.1.18 /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorEventsEnable

**Type:** string

**Description:** The required *SensorsEventsEnable* parameter shall enable reporting of events for the indicated sensor. On read of this parameter a CSV of pairs of supported event names and a boolean value with ("1") indicating that the corresponding event is to be reported. On write a CSV of an event name and boolean value indicating the new enable/disable setting for a supported named event shall be accepted. Sensor events are reported by the SensorEvents XML document provided by the *[/UPnP/SensorMgt/SensorEvents](#)* parameter.

The following *SensorEventsEnable* names are defined:

**Table A.2 — SensorEventEnable parameter allowed values**

| SensorEventEnable Name | Description |
|---|---|
| *SOAPDataAvailableEnable* | Controls reporting of SOAPDataAvailable events indicating sensor data is available for reading by the ReadSensor() action of the SensorTransportGeneric service. |
| *SOAPDataOverrunEnable* | Controls reporting of SOAPDataOverrun events indicating the sensor has discarded data which was not read by any SOAP client. |
| *TransportDataAvailableEnable* | Controls reporting of TransportDataAvailable events indicating a sensor with one or more active transport connections has received data. |
| *TransportDataOverrunEnable* | Controls reporting of TransportDataOverrun events indicating a sensor with one or more active transport connections has discarded received data it was unable to forward to transport clients. |
| *TransportConnectionErrorEnable* | Controls reporting of TransportConnectionError events indicating a sensor with one or more active transport connections has detected a connection error. |
| *SensorAvailabilityEnable* | Controls reporting of SensorAvailable and SensorUnavailable events indicating a previously reported sensor has become accessible or inaccessible. |

The following example shows a sample SensorEventsEnable parameter and changing one of the reported event settings:

```
Request:
GetValues(
   "<?xml version="1.0" encoding="UTF-8"?>
    <ContentPathList xmlns="urn:schemas-upnp-org:dm:cms"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance
      xsi:schemaLocation="urn:schemas-upnp-org:dm:cms
                            http://www.upnp.org/schemas/dm/cms.xsd">
       <ContentPath>
         /UPnP/SensorMgt/SensorCollections/1/Sensors/1/SensorEventsEnable
      </ContentPath>
    </ContentPathList>" )

Response:
GetValues(
   "<?xml version="1.0" encoding="UTF-8"?>
    <ParameterValueList xmlns="urn:schemas-upnp-org:dm:cms"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance
      xsi:schemaLocation="urn:schemas-upnp-org:dm:cms
                            http://www.upnp.org/schemas/dm/cms.xsd">
      <Parameter>
         <ParameterPath>
            /UPnP/SensorMgt/SensorCollections/1/Sensors/1/SensorEventsEnable
         </ParameterPath>
         <Value>
            SOAPDataAvailableEnable,0,SOAPDataOverrunEnable,0,
            TransportDataAvailableEnable,0,TransportDataOverrunEnable,0,
            TransportConnectionErrorEnable,0
         </Value>
      </Parameter>
   </ParameterValueList>"  )
```

A SOAP client which requests notification of SensorDataAvailable events updates this parameter as follows:

```
Request:
SetValues(
   "<?xml version="1.0" encoding="UTF-8"?>
    <ParameterValueList xmlns="urn:schemas-upnp-org:dm:cms"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance
      xsi:schemaLocation="urn:schemas-upnp-org:dm:cms
                          http://www.upnp.org/schemas/dm/cms.xsd">
      <Parameter>
         <ParameterPath>
            /UPnP/SensorMgt/SensorCollections/1/Sensors/1/SensorEventsEnable
         </ParameterPath>
         <Value>
            SOAPDataAvailableEnable,1
         </Value>
      </Parameter>
   </ParameterValueList>"  )

Response:
SetValues(
   "ChangesCommitted")
```

### A.1.1.19 /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorsRelatedNumberOfEntries

**Type:** int
**Description:** The conditionally required *SensorsRelatedNumberOfEntries* *parameter* shall provide the number of *SensorsRelated* MultiInstance node entries. This *parameter* shall be implemented if the *SensorsRelated* MultiInstance node is supported.

### A.1.1.20 /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorGroupsNumberOfEntries

**Type:** int
**Description:** The conditionally required *SensorsGroupsOfEntries* *parameter* shall provide the number of *SensorGroups* MultiInstance node entries. This *parameter* shall be implemented if the *SensorGroups* MultiInstance node is supported.

### A.1.1.21 /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorDefaultPermissionsNumberOfEntries

**Type:** int
**Description:** The conditionally required *SensorsPermissionNumberOfEntries* *parameter* shall provide the number of *SensorPermissions* MultiInstance node entries. . This *parameter* shall be implemented if the *SensorDefaultPermissions* MultiInstance node is supported

### A.1.1.22 /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorsRelated/

**Type:** MultiInstance
**Description:** The allowed *SensorsRelated* MultiInstance node includes zero or more *SensorPath* parameter nodes identifying other sensors which have an operational relationship to this sensor. Note: This differs from the *SensorGroups* node where sensors are grouped for classification and device protection.

### A.1.1.23 /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorsRelated/#/SensorPath

**Type:** string
**Description:** The required *SensorPath* parameter provides a relative path to a sensor operationally related to this sensor. This path shall start at the *SensorCollections* MultiInstance node and shall end at the *Sensors/#* instance node identifying the sensor being referenced.

### A.1.1.24 /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorGroups/

**Type:** MultiInstance
**Description:** The conditionally required *SensorGroups* MultiInstance node includes zero or more *SensorGroup* parameter nodes identifying SensorGroup(s) this sensor is participating in.

The *SensorGroups* MultiInstance node shall be implemented when the DeviceProtection feature is supported.

Note: See IoT Management and Control Architecture Overview [10], subclause 4.1.9 "Sensor Protection Model" for further details on usage of SensorGroup(s) for DeviceProtection.

### A.1.1.25 /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorGroups/#/SensorGroup

**Type:** string

**Description:** The required *SensorGroup* parameter identifies an individual SensorGroup the corresponding Sensor is participating in. An instance of the *SensorGroup* node shall be included for each sensor group the sensor is a member of.

### A.1.1.26 /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorDefaultPermissions/

**Type:** MultiInstance

**Description:** The conditionally allowed *SensorPermissions* MultiInstance node lists sensor permissions for default DeviceProtection roles. This node may be implemented if the IoT Management and Control device provides DeviceProtection support and is otherwise prohibited.

### A.1.1.27 /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorPermissions/#/SensorDefaultRole

**Type:** string

**Description:** The required *SensorDefaultRole* parameter identifies a default DeviceProtection role. The allowed values for this parameter are the default DeviceProtection roles: “*Public*” or “*Basic*”.

### A.1.1.28 /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorPermissions/#/SensorDefaultPermissions

**Type:** CSV(string)

**Description:** The required *SensorDefaultPermissions* parameter lists SensorPermissions assigned to the corresponding default DeviceProtection role identified by the *SensorDefaultRole* parameter. This parameter shall be supported if the IoT Management and Control device provides DeviceProtection support. The following allowed values are defined.

**Table A.3 — Sensor Permissions**

| Permission | Description |
|---|---|
| *smgt:ReadSensor* | A control point is permitted to issue *ReadSensor()* actions to the corresponding Sensor. |
| *smgt:WriteSensor* | A control point is permitted to issue *WriteSensor()* actions to the corresponding Sensor. |
| *smgt:ConnectSensor* | A control point is permitted to issue *ConnectSensor()* and *DisconnectSensor()* actions to the corresponding Sensor. |
| *smgt:CommandSensor* | A control point is permitted to modify IoT Management and Control properties in the DataModel. |
| *smgt:ViewSensor* | A control point is permitted to view IoT Management and Control properties for this Sensor. |

### A.1.1.29 /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorSpecific/

**Type:** SingleInstance

**Description**: The required *SensorSpecific* SingleInstance node provides ecosystem specific Sensor information. The allowed and required children of this node depend on the specific type of sensor as indicated by the *SensorType* parameter. Values prefixed by “smgt:” are defined within this specification. See the sections describing specific classes of sensor collection(s) for required and allowed nodes for specific sensor types.

### A.1.1.30 /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorsURNsNumberOfEntries

**Type:** int
**Description:** The required *SensorURNsNumberOfEntries* parameter shall provide the number of *SensorURNs* MultiInstance node entries.

### A.1.1.31 /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorURNs/

**Type:** MultiInstance
**Description:** The required *SensorURNs* MultiInstance node lists URN values for sets of DataItem(s) this sensor provides (or actuator accepts).

### A.1.1.32 /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorURNs/#/SensorURN

**Type:** string
**Description:** The required *SensorURN* parameter node identifies the URN value for each set of sensor DataItem(s). See the sections describing specific classes of sensor collection(s) for required and allowed URN values for sensor types.

### A.1.1.33 /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorURNs/DataItemsNumberOfEntries

**Type:** int
**Description:** The required *DataItemsNumberOfEntries* parameter shall provide the number of *DataItems* MultiInstance node entries.

### A.1.1.34 /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorURNs/#/DataItems/

**Type:** MultiInstance
**Description:** The required *DataItems* MultiInstance node identifies sets of DataItem(s) supported for the corresponding *SensorURN* parameter value.

### A.1.1.35 /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorURNs/#/DataItems/#/Name

**Type:** MultiInstance
**Description:** The required *Name* parameter identifies the name of an individual DataItem. See IoT Management and Control Architecture Overview [10] subclause 4.3, "DataItem semantics" for further information.

### A.1.1.36 /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorURNs/#/DataItems/#/Type

**Type:** MultiInstance
**Description:** The required *Type parameter* identifies the type of an individual DataItem described by the corresponding *SensorURN parameter* value. See IoT Management and Control Architecture Overview [10] subclause 4.3, "DataItem semantics" for further information.

### A.1.1.37 /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorURNs/#/DataItems/#/Encoding

**Type:** MultiInstance
**Description:** The required *Encoding* parameter identifies the encoding of an individual *DataItem* described by the corresponding *SensorURN* parameter value. See IoT Management and Control Architecture Overview [10] subclause 4.3, "DataItem semantics" for further information.

### A.1.1.38 /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorURNs/#/DataItems/#/Description

**Type:** MultiInstance

**Description:** The required *Description parameter* shall provide a string containing an XML document which describes the named DataItem or an empty string if no DataItem Description document is available. This XML document shall comply with the DataItem Description schema. See IoT Management and Control Architecture Overview [10] subclause 4.3, "DataItem semantics" for further information.

### A.1.1.39 /UPnP/SensorMgt/SensorGroupSets/

**Type:** MultiInstance

**Description:** The conditionally required *SensorGroupSets* MultiInstance node identifies zero or more *SensorGroupSets* definition. As SensorGroupSet can be made by ControlPoint after initial setting, this node and all the parameters under this node are writable. SensorGroupSet can contain any combination of SensorCollections or Sensors.

### A.1.1.40 /UPnP/SensorMgt/SensorGroupSets/#/GroupSetID

**Type:** string

**Description:** The required *GroupSetID parameter* provides a unique identifier for the purpose of accessing this SensorGroupSet.

### A.1.1.41 /UPnP/SensorMgt/SensorGroupSets/#/GroupSetType

**Type:** string

**Description:** The required *GroupSetType* parameter value describes the type of *SensorGroupSet*. *GroupSetType* identifier is *sgmt-gst*.

**Table A.4 — GroupSetType values**

| GroupSetType value | Description |
|---|---|
| *smgt-gst:access-control* | GroupSet intended to define access control of this group members. The *SensorPermissions* MultiInstance node (see description in A.1.1.26) for this group will be placed under *GroupSetSpecific* Instance node. |
| *smgt-gst:control-group* | *GroupSet* intended to define a control group in which members will be controlled as a single entity.<br>No *GroupSetSpecific* instance is required if all members have the same control scheme defined in the *SensorURNs*.<br>If it is desired to limit only a few *SensorURNs* for control, they will be listed under *GroupSetSpecific* instance. A SensorURNsNumberOfEntries parameter must be included indicating the number of *SensorURNs* listed. Then, a MultiInstance node *SensorURNs* with all *SensorURN* parameters is included.<br>If no *GroupSetSpecific* instance is defined, the common set of *SensorURNs* defined between the members will be used for control. |
| *smgt-gst:group-represent* | GroupSet intended to define representing sensor that will represent other sensor members where similar sensors are located together. *SensorID* of that representing sensor will be described under *GroupSetSpecific* Instance. |

### A.1.1.42 /UPnP/SensorMgt/SensorGroupSets/#/GroupSetFriendlyName

**Type:** string

**Description:** The required *GroupSetFriendlyName* parameter provides a string identifying the *SensorGroupSet* name suitable for end-user display.

### A.1.1.43 /UPnP/SensorMgt/SensorGroupSets/#/GroupSetSpecific/

**Type:** SingleInstance

**Description:** The conditionally required *GroupSetSpecific* SingleInstance node provides ecosystem specific *SensorGroupSet* information. The child nodes of the *GroupSetSpecific* node depend on the *GroupSetType* parameter value.

### A.1.1.44 /UPnP/SensorMgt/SensorGroupSets/#/MembersNumberOfEntries

**Type:** int

**Description:** The required *MembersNumberOfEntries* parameter shall provide the number of *Member* MultiInstance node entries.

### A. 1.1.45 /UPnP/SensorMgt/SensorGroupSets/#/Members/#/

**Type:** MultiInstance

**Description:** The required *Members* MultiInstance node identifies zero or more members of this SensorGroupSet. As SensorGroupSet can contain any combination of SensorCollection or Sensor, member can be SensorCollection or Sensor or both.

### A.1.1.46 /UPnP/SensorMgt/SensorGroupSets/#/MemberID

**Type:** string

**Description:** The required *MemberID* *parameter* provides a referencing identifier of this member. *CollectionID* or *SensorID* of the member is used here.

### A.1.1.47 UPnP/SensorMgt/SensorCollections/#/ ParticipatingGroupSetID, UPnP/SensorMgt/SensorCollections/#/ Sensors/#ParticipatingGroupSetID

**Type:** string

**Description:** The required *ParticipatingGroupSetID* *parameter* under SensorCollection or Sensor provides a referencing identifier to the participating SensorGroupSet.

## A.2 Script Settings

Predefined settings for a group of sensors are allowed by the use of the Data Model tree namely *ScriptSettings*. Using *Settings* parameters it is possible to predefine *DataRecords* to be applied to a group of sensors. For example, it is possible to predefine lighting settings for movies in your living-room.

**Table A.5 — ScriptSettings Parameters for predefined Settings**

| Name | Type | Acc | Req | Reference | EOC | Ver |
|---|---|---|---|---|---|---|
| | | | | A.1.1.29 | | |
| /UPnP/ScriptSettings/ | SingleInstance | RO | A | | | 0 |
| SettingsNumberOfEntries | int | RO | R | | | 0 |
| | | | | | | |
| /UPnP/ScriptSettings/Permissions/#/ | MultiInstance | RO | CA | | | |
| DefaultRole | string | RO | R | | | |
| SettingsDefaultPermissions | CSV (string) | RO | R | | | |
| ScriptsDefaultPermissions | CSV (string) | RO | R | | | |
| | | | | | | |
| /UPnP/ScriptSettings/Settings/#/ | MultiInstance | RW | CR | | | 0 |
| SettingID | string | RW | R | | | |
| SettingDescription | string | RW | R | | | |
| ConfigsNumberOfEntries | int | RW | R | | | |
| | | | | | | |
| /UPnP/ ScriptSettings/Settings/#/Configs/#/ | MultiInstance | RW | R | | | 0 |
| DeviceID | string | RW | R | | | |
| SensorURN | string | RW | R | | | 0 |
| DataRecords | string | RW | R | | | 0 |
| SettingPermissionsNumberOfEntries | int | RW | A | | | |
| | | | | | | |
| /UPnP/ScriptSettings/Settings/#/SettingPermissions/#/ | MultiInstance | RW | CA | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| SettingDefaultRole | string | RW | R | | | |
| SettingDefaultPermissions | CSV (string) | RW | R | | | |
| | | | | | | |

### A.2.1 ScriptSettings General Parameters

#### A.2.1.1 /UPnP/ScriptSettings/

**Type:** SingleInstance
**Description:** The conditionally required *ScriptSettings* SingleInstance node shall be included for pre-defined Settings support.

#### A.2.1.2 /UPnP/ScriptSettings/SettingsNumberOfEntries

**Type:** int
**Description:** The conditionally required *SettingsNumberOfEntries* indicates the number of Settings available for this *SensorManagement* device.

#### A.2.1.3 /UPnP/ScriptSettings/SettingsScriptsPermissions

**Type:** MultiInstance

**Description:** The conditionally allowed *SettingsScriptsPermissions* MultiInstance node lists permissions for the addition of Settings and Scripts for default *DeviceProtection* roles. This node may be implemented if the SensorManagement device provides *DeviceProtection* support and is otherwise prohibited.

#### A.2.1.4 /UPnP/ScriptSettings/SettingsScriptsPermissions/#/DefaultRole

**Type:** string
**Description:** The required *DefaultRole* parameter identifies a default DeviceProtection role. The allowed values for this parameter are the default DeviceProtection roles: **"***Public***"** or **"***Basic***".** The default role **"***Admin***"** does not need to be listed and has permission to all operations.

#### A.2.1.5 /UPnP/ScriptSettings/SettingsScriptsPermissions/#/SettingsDefaultPermissions

**Type:** CSV (string)

**Description:** The required *SettingsDefaultPermissions* parameter lists SettingsScriptsPermissions assigned to the corresponding default DeviceProtection role identified by the *DefaultRole* parameter for *Settings*. This parameter shall be supported if the SensorManagement device provides *DeviceProtection* support. The allowed values are defined in Table A.6.

**Table A.6 — Default permission settings**

| Permission | Description |
|---|---|
| *sset:CreateSetting* | A control point is permitted to create a node in the *Settings* multiinstance node in the DataModel. |
| *sset:ChangeAnySetting* | A control point is permitted to change any parameter values of any node in the *Settings* multiinstance node in the DataModel. |
| *sset:DeleteAnySetting* | A control point is permitted to remove any node in the *Settings* multiinstance node in the DataModel. |
| *sset:ViewAnySettings* | A control point is permitted to view properties and parameter of any node in the *Settings* multiinstance node in the DataModel. |

### A.2.1.6 /UPnP/ScriptSettings/SettingsScriptsPermissions/#/ScriptsDefaultPermissions

**Type:** CSV (string)

**Description:** The required *ScriptsDefaultPermissions* parameter lists SettingsScriptsPermissions assigned to the corresponding default DeviceProtection role identified by the *DefaultRole* parameter for *Scripts*. This parameter shall be supported if the SensorManagement device provides *DeviceProtection* support. The alloed values are defined in Table A.7.

**Table A.7 — Default script permission settings**

| Permission | Description |
|---|---|
| *sset:CreateScript* | A control point is permitted to create a node in the *Scripts* multiinstance node in the DataModel. |
| *sset:ChangeAnyScript* | A control point is permitted to change any parameter values of any node in the *Scripts* multiinstance node in the DataModel. |
| *sset:DeleteAnyScript* | A control point is permitted to remove any node in the *Scripts* multiinstance node in the DataModel. |
| *sset:ViewAnyScripts* | A control point is permitted to view any properties and parameter of any node in the *Scripts* multiinstance node in the DataModel. |

### A.2.1.7 /UPnP/ScriptSettings/Settings/

**Type:** MultiInstance

**Description:** The conditionally required *Settings* MultiInstance node identifies zero or more pre-defined settings for the *IoT Management and Control Device*.

### A.2.1.8 /UPnP/ScriptSettings/Settings/#/SettingID

**Type:** string

**Description:** The conditionally required *SettingID* parameter provides a unique identifier for the purpose of applying this *Setting* based on a *Script*.

### A.2.1.9 /UPnP/ScriptSettings/Settings/#/SettingDescription

**Type:** string

**Description:** The conditionally required *SettingDescription* provides a user-friendly description about this pre-defined *Setting*.

### A.2.1.10 /UPnP/ScriptSettings/Settings/#/ConfigsNumberOfEntries

**Type:** string

**Description:** The conditionally required *ConfigsNumberOfEntries* parameter shall provide the number of *Configs* MultiInstance nodes.

### A.2.1.11 /UPnP/ScriptSettings/Settings/#/Configs/#/

**Type:** MultiInstance

**Description:** The conditionally required *Configs* MultiInstance node identifies one or more set of configuration parameters that are used in the *Setting*.

### A.2.1.12 /UPnP/ScriptSettings/Settings/#/Configs/#/DeviceID

**Type:** string

**Description:** The conditionally required *DeviceID* parameter identifies which *SensorID* or *GroupSetID* the following *DataRecords* must be applied.

### A.2.1.13 /UPnP/ScriptSettings/Settings/#/Configs/#/SensorURN

**Type:** string
**Description:** The conditionally required *SensorURN* parameter identifies which *SensorURN* of the *SensorID* or *GroupSetID* the following *DataRecords* must be applied.

### A.2.1.14 /UPnP/ScriptSettings/Settings/#/Configs/#/DataRecords

**Type:** string
**Description:** The conditionally required *DataRecords* parameter have the DataRecords XML to be applied on the desired *SensorID* **or** *GroupSetID* (See Section 5.4.5 of IoT Management and Control TransportGeneric Service).

### A.2.1.15 /UPnP/ScriptSettings/Settings/#/SettingPermissionsNumberOfEntries

**Type:** int

**Description:** The allowed *SettingPermissionsNumberOfEntries* indicates the number of *SettingPermissions* available for this *Setting*. This parameter shall be implemented if the *SettingPermissions* MultiInstance node is supported

### A.2.1.16 /UPnP/ScriptSettings/Settings/#/SettingPermissions/#/

**Type:** MultiInstance

**Description:** The conditionally allowed *SettingPermissions* MultiInstance node lists permissions for changing the Setting for default *DeviceProtection* roles. This node may be implemented if the SensorManagement device provides *DeviceProtection* support and is otherwise prohibited.

### A.2.1.17 /UPnP/ScriptSettings/Settings/#/SettingPermissions/#/SettingDefaultRole

**Type:** string

**Description:** The required *SettingDefaultRole* parameter identifies a default DeviceProtection role. The allowed values for this parameter are the default DeviceProtection roles: **"***Public***" or "***Basic***".** The default role **"***Admin***"** does not need to be listed and has permission to all operations.

### A.2.1.18 /UPnP/ScriptSettings/Settings/#/SettingPermissions/#/SettingDefaultPermissions

**Type:** CSV (string)

**Description:** The required *SettingDefaultPermissions* parameter lists permissions assigned to the corresponding default DeviceProtection role identified by the *SettingDefaultRole* parameter for the *Setting.* This parameter shall be supported if the SensorManagement device provides *DeviceProtection* support. The allowed values are defined in Table A.8.

**Table A.8 — Default permissions settings**

| Permission | Description |
|---|---|
| *sset:ChangeSetting* | A control point is permitted to change any parameter values of the parent *Setting* node in the DataModel. |
| *sset:DeleteSetting* | A control point is permitted to remove the parent *Setting* node in the DataModel. |
| *sset:ViewSetting* | A control point is permitted to view properties and parameter of the parent *Setting* node in the DataModel. If a node is allowed to *sset:ChangeSetting* and *sset:*DeleteSetting it is automatically allowed to *sset:ViewSetting*. |

## A.3 Scripts

Scripts are used to apply Settings based on predefined Conditions. For example, it is possible to create a Script that turn off all lights of the living room after a presence Sensor indicates that nobody is there. The parameters to achieve this functionality are defined in Table A.9.

**Table A.9 —ScriptSettings Parameters for Scripts**

| Name | Type | Acc | Req | Reference | EOC | Ver |
|---|---|---|---|---|---|---|
| | | | | A.1.1.29 | | |
| /UPnP/ScriptSettings/ | SingleInstance | RO | A | | | 0 |
| SettingsNumberOfEntries | int | RO | R | | | 0 |
| ScriptsNumberOfEntries | int | RO | R | | | |
| | | | | | | |
| /UPnP/ScriptSettings/Scripts/#/ | MultiInstance | RW | CR | | | 0 |
| ScriptID | string | RW | R | | | |
| ScriptDescription | string | RW | R | | | |
| ScriptAttributes | CSV(string) | RW | R | | | |
| ScriptStatus | string | RW | R | | | |
| Condition | string | RW | R | | | 0 |
| Then | CSV(string) | RW | CR | | | |
| ScriptPermissionsNumberOfEntries | int | RW | A | | | |
| | | | | | | |
| /UPnP/ScriptSettings/Scripts/#/ScriptPermissions/#/ | MultiInstance | RW | CA | | | |
| ScriptDefaultRole | string | RW | R | | | |
| ScriptDefaultPermissions | CSV (string) | RW | R | | | |
| | | | | | | |

### A.3.1 ScriptSettings General Parameters for Scripts

#### A.3.1.1 /UPnP/ScriptSettings/ScriptsNumberOfEntries

**Type:** int

**Description:** The conditionally required *ScriptsNumberOfEntries* indicates the number of *Scripts* available for this *SensorManagement* device.

#### A.3.1.2 /UPnP/ScriptSettings/Scripts/

**Type:** MulitInstance

**Description:** The conditionally required *Scripts* MultiInstance node identifies zero or more set of scripts available for this *SensorManagement* device.

#### A.3.1.3 /UPnP/ScriptSettings/Scripts/#/ScriptID

**Type:** string

**Description:** The conditionally required *ScriptID* parameter provides a unique identifier for ths *SensorManagement* device.

#### A.3.1.4 /UPnP/ScriptSettings/Scripts/#/ScriptDescription

**Type:** string

**Description:** The conditionally required *ScriptDescription* parameter provides a user-friendly description of this script.

#### A.3.1.5 /UPnP/ScriptSettings/Scripts/#/ScriptAttributes

**Type:** CSV (string)

**Description:** The conditionally required *ScriptAttributes* parameter provides a list of zero or more attributes that must be applied to the script. If no attribute is necessary for the *Script*, this parameter must be empty.The attributes that are supported are defined in Table A.10.

**Table A.10 — Script Attibutes**

| Script Attribute | Description |
|---|---|
| delay | Indicates that the “Then” will fire after a specified delay if “Condition” is still true |
| repeats | Indicates the number of times “Then” must be applied if “Condition” is still true. The keyword “infinite” can be used. |
| sched-start | Indicates that “Then” can only be activated after sched-start if “Condition” is still true |
| date-time-range | Indicates that “Then” can only be activated into this range if “Condition” is still true |
| date-time-expires | Indicates that the “Script” must be deactivated after date-time-expires |
| delay-expires | Indicates that the “Script” must be deactivated after delay-expires. |
| periodic | Indicates a periodic time to keep executing the “Script”. |
| moderation-time | Indicates a “pause” period to execute the “Script” again. |

The syntax is formally defined using EBNF. The syntax is described as follow:

```
delay                ::= duration
delay-expires        ::= duration
moderation-time      ::= duration
periodic             :: duration
date-time-expires    ::= date-time
sched-start          ::= date-time | day-of-yr-time | named-day-time | T-labeled-time
| 'NOW'
date-time-range      ::= date-time '/' date-time
duration             ::= 'P' [n 'D'] time


date-time            ::= yyyy '-' mm '-' dd T-labeled-time
day-of-yr-time       ::= mm '-' dd T-labeled-time
named-day-time       ::= named-day T-labeled-time
T-labeled-time       ::= 'T' time [zone]
time                 ::= HH ':' MM ':' SS
zone ::= 'Z'|(('+'|'-') HH ':' MM)
month-day ::= mm '-' dd
named-day ::= 'MON'|'TUE'|'WED'|'THU'|'FRI'|'SAT'|'SUN'| 'MON-FRI'|'MON-SAT'
n ::= 1*DIGIT (* non-negative integer *)
yyyy ::= 4DIGIT (* 0001-9999 *)
mm ::= 2DIGIT (* 01-12 *)
dd ::= 2DIGIT (* 01-28, 01-29, 01-30, 01-31 based on month/year *)
HH ::= 2DIGIT (* 00-23 *)
MM ::= 2DIGIT (* 00-59 *)
SS ::= 2DIGIT (* 00-59 *)
```

### A.3.1.6 /UPnP/ScriptSettings/Scripts/#/ScriptStatus

**Type:** string

**Description:** The conditionally required *ScriptStatus* parameter provides the current status of the Script. The values that are supported are listed in Table A.11.

**Table A.11 — Script Status values**

| Script Status | Description |
|---|---|
| activated | Indicates that the Script is activated, which means that Conditions will be evaluated. |
| deactivated | Indicates that the Script is deactivated, which means that Conditions will NOT be evaluated. |
| running | Indicates that the Script is activated and is evaluating the conditions OR applying the Settings. This ScriptStatus can NOT be changed by the user (or control point). |
| error | Indicates that the last time the script was executed an error occurred, and the script must be reviewed. |

The *ScriptStatus* *activated*, *deactivated* or *error* are controlled by the user (or control point). The *ScriptStatus* *running* are not controlled by the user (or control point), therefore, it is necessary to wait for its conclusion.
The behavior of these *ScriptStatus* are illustrated in the state machine described in **Figure A.1**.

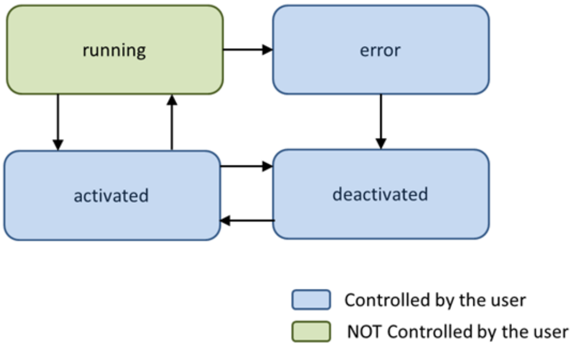

**Figure A.1 — Script status state machine**

### A.3.1.7 /UPnP/ScriptSettings/Scripts/#/Condition

**Type:** string

**Description:** The conditionally required *Condition* parameter must follow a specific syntax, as described in Section A.1.3, “Condition EBNF”.

### A.3.1.8 /UPnP/ScriptSettings/Scripts/#/Then

**Type:** CSV (string)

**Description:** The conditionally required *Then* parameter lists one or more *Settings* to be applied**.** *Settings* must be applied in the order that they are listed. *Settings* must be referenced by its *SettingsID*.
If the attribute *repeats* is not defined in the *ScriptAttributes* parameter, the script must be executed just one (1) time and go back to *deactivated* status.

### A.3.1.9 /UPnP/ScriptSettings/Scripts/#/ScriptPermissionsNumberOfEntries

**Type:** int

**Description:** The allowed *ScriptPermissionsNumberOfEntries* indicates the number of *ScriptPermissions* available for this *Script*. This parameter shall be implemented if the *ScriptPermissions* MultiInstance node is supported.

### A.3.1.10 /UPnP/ScriptSettings/Scripts/#/ScriptPermissions

**Type:** MultiInstance

**Description: The conditionally allowed** *ScripPermissions* MultiInstance node lists permissions for changing the parent *Script* for default *DeviceProtection* roles. This node may be implemented if the SensorManagement device provides *DeviceProtection* support and is otherwise prohibited.

### A.3.1.11 /UPnP/ScriptSettings/Scripts/#/ScriptPermissions/#/ScriptDefaultRole

**Type:** string

**Description:** The required *ScriptDefaultRole* parameter identifies a default DeviceProtection role. The allowed values for this parameter are the default DeviceProtection roles: “*Public*” or “*Basic*”. The default role “*Admin*” does not need to be listed and has permission to all operations

### A.3.1.12 /UPnP/ScriptSettings/Scripts/#/ScriptPermissions/#/ScriptDefaultPermissions

**Type:** CSV (string)

**Description:** The required *ScriptDefaultPermissions* parameter lists permissions assigned to the corresponding default DeviceProtection role identified by the ScriptDefaultRole parameter for the *Script*. This parameter shall be supported if the SensorManagement device provides *DeviceProtection* support. The allowed values are defined in Table A.12.

**Table A.12 — Script default permissions**

| Permission | Description |
|---|---|
| *sset:ChangeScript* | A control point is permitted to change any parameter values of the parent *Script* node in the DataModel. |
| *sset:DeleteScript* | A control point is permitted to remove the parent *Script* node in the DataModel. |
| *sset:ViewScript* | A control point is permitted to view properties and parameter of the parent *Script* node in the DataModel. If a node is allowed to *sset:ChangeScript and sset:DeleteScript it is automatically allowed to sset:ViewScript.* |

### A.3.2 Condition EBNF

The Condition parameter string syntax is described here formally using EBNF as described in Section 4.3.1, "Extended Backus-Naur Form”. A SensorManagement Data Model service implementation MUST support the entire syntax as described below

```
condCrit            ::= condExp|true

condExp             ::= relExp|eventExp|

                     condExp wChar+ logOp wChar+ condExp |

                     '(' wChar* condExp wChar* ')'

logOp               ::= 'and'|'or'

eventExp            ::= (* event property as defined in Section A.1.5 *)

relExp              ::= property wChar+ binOp wChar+ quotedVal|

                     property wChar+ existsOp wChar+ boolVal

binOp               ::= relOp|stringOp

relOp               ::= '='|'!='|'<'|'<='|'>'|'>='

stringOp            ::= 'contains'|'doesNotContain'|'derivedfrom'|

                     'startsWith'|'derivedFrom'

existsOp            ::= 'exists'
```

```
boolVal             ::= 'true'|'false'
quotedVal           ::= dQuote escapedQuote dQuote
wChar               ::= space|hTab|lineFeed|vTab|formFeed|return
property            ::= (* script property name as defined in Section A.1.4 *)
escapedQuote        ::= (* double-quote escaped string as defined in
                         Section TBD *)
hTab                ::= (* UTF-8 code 0x09, horizontal tab character *)
lineFeed            ::= (* UTF-8 code 0x0A, line feed character *)
vTab                ::= (* UTF-8 code 0x0B, vertical tab character *)
formFeed            ::= (* UTF-8 code 0x0C, form feed character *)
return              ::= (* UTF-8 code 0x0D, carriage return character *)
space               ::= ' '
                         (* UTF-8 code 0x20, space character *)
dQuote              ::= '"'
                         (* UTF-8 code 0x22, double quote character *)
asterisk            ::= '*'
                         (* UTF-8 code 0x2A, asterisk character *)
```

### A.3.2.1 Condition Examples

1) The Condition is true if the *Sensor* with *SensorID* “Sensor01” contains in its type the string “refrigerator” and if any of its *DataItems* with the Name “Temperature” has the value of 30.
   - smgt-st:Sensor01@SensorType contains “refrigerator” and smgt-st:Sensor01@SensorURNs/#/DataItems/#/Temperature/Value = “30”

2) The Condition is true when the event SensorAdded is fired and any Sensor contains in its type the string “refrigerator”.

   - smgt-evt:SensorAdded and smgt-st:*@SensorType contains “refrigerator”

### A.3.3 Script Properties

A Script property represents a Data Model parameter. These properties are used to check values for Conditions into a Script.

A property has the following general template:

- Property ::= [property-identifier]:[identifier-value]@[DataPath-from-BaseNode]

The [property-identifier] shall use the values defined in Table A.13.

**Table A.13 — Property-identifier values for Scripts**

| Property Name | [property-identifier] | Base Node |
|---|---|---|
| Sensor Management | *smgt* | */UPnP/SensorMgt/* |
| Sensor Collection | *smgt-sct* | /UPnP/SensorMgt/SensorCollections/#/ |
| Sensor | *smgt-st* | /UPnP/SensorMgt/SensorCollections/#/Sensors/#/ |
| Sensor URN | *smgt-surn* | /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorURNs/#/ |
| Sensor Script | *sset-scrt* | /UPnP/ScriptSettings/Scripts/#/ |

If the property-identifier is smgt, the following template must be used:

- Property ::= smgt:[DataPath-from-BaseNode]

*DataPath* is the path from the *Base Node* to a parameter *Node* which might include the wild-chars # instead of table *Instances* numbers, however these can never be mixed within the same path. Examples of *DataPaths* for smgt are:

- SensorCollections/#/CollectionID
  - Can be used to access all values of CollectionID
- SensorCollections/1/CollectionID
  - Can be used to access the value of CollectionID of *Instance* number 1.
- SensorCollections/2/
  - Can be used to check if the *Instance* number 2 exists.

The [identifier-value] is used to simplify the access for a specific parameter *Node* based on the *DataPath* and an identification (ID) parameter. For example, a Property can access nodes from a specific Collection based on its CollectionID value, or access a specific Sensor based on its SensorID value. The [identifier-value] for each [property-identifier] shall use the parameter values described in Table A.14.

**Table A.14 — property-identifier values for collections.**

| Property Name | [property-identifier] | [identifier-value] |
|---|---|---|
| Sensor Collection | smgt-sct | CollectionID value |
| Sensor | smgt-st | SensorID value |
| Sensor URN | smgt-surn | SensorURN value |
| Sensor Script | sset-scrt | ScriptID value |

The [identifier-value] might include the wild-chars '*'.

Therefore, for each propert-identifier the Property template is the following:

- smgt-sct:[CollectionID]@ [DataPath-from-BaseNode]
- smgt-st:[SensorID]@ [DataPath-from-BaseNode]
- smgt-surn:[SensorURN]@ [DataPath-from-BaseNode]
- sset-scrt:[ScriptID]@ScriptStatus
  - The sset-scrt property-identifier must only access the ScriptStatus parameter.

The value of a *DataItem* can be accessed through two methods by the use of a virtual paramenter namely *Value*.

The first method is based on the *DataItem Name* and wild-chars #, as the following template:

- smgt-surn:[SensorURN] @DataItems/#/[Name]/Value
- smgt-st:[SensorID]@SensorURNs/#/DataItems/#/[Name]/Value
- smgt-sct: [CollectionID]@Sensors/#/SensorURNs/#/DataItems/#/[Name]/Value
- smgt:SensorCollections/#/Sensors/#/SensorURNs/#/DataItems/#/[Name]/Value

Where [Name] is the name of the *DataItem Name*.

*DataItem Values* can be also accessed based on the *DataItem Instance*, as the following template:

- smgt-surn:[SensorURN] @DataItems/1…/Value
- smgt-st:[SensorID]@SensorURNs/1…/DataItems/1…/Value
- smgt-sct: [CollectionID]@Sensors/1…/SensorURNs/1…/DataItems/1…/Value
- smgt:SensorCollections/1…/Sensors/1…/SensorURNs/1…/DataItems/1…/Value

#### A.3.3.1 Property Examples

Here are presented a list of property examples:

- *smgt-sct:Collection01@CollectionType*
  - Points to the CollectionType of Collection01
- *smgt-sct:Collection01@Sensors/#/SensorID*
  - It is the Set of SensorIDs into Collection01
- *smgt-sct:Collection*@Sensors/1/SensorID*
  - It is the Set of SensorIDs of first instances of all Collections starting with “Collection”
- *smgt-sct:Collection01@Sensors/1/SensorID*
  - Points to the SensorID value of the first instance of Collection01

- *smgt-st:Sensor01@SensorType*
  - Points to the SensorType value of Sensor01
- *smgt-st:Sensor01@SensorGroups/#/SensorGroup*
  - It is the Set of SensorGroups of Sensor01
- *smgt-st:Sensor*@SensorGroups/#/SensorGroup*
  - It is the Set of SensorGroups of all Sensors with SensorID starting with “Sensor”
- *smgt-st:Sensor01@SensorURNs/#/DataItems/#/Name*
  - It is the Set of Names of all DataItems of Sensor01.
- *smgt-st:Sensor01@SensorURNs/1/DataItems/1/ Value*
  - It is the Value of the first DataItem Instance of the first SensorURN Instance

- *smgt-surn:urn:upnp-org:smgt-surn:light-controls:AcmeSensorsCorp-com:AcmeIntegratedController:LightsCorp:rf217acrs:control@DataItems/#/Name*
  - It is the Name of all DataItems of SensorURN.
- *smgt-surn:urn:upnp-org:smgt-surn:light-controls:AcmeSensorsCorp-com:AcmeIntegratedController:LightsCorp:rf217acrs:control@DataItems/1/ Value*
  - It is the Set of Values of the first DataItem Instance of all SensorURNs

- *sset-scrt:Script01@ScriptStatus*
  - Points to the ScriptStatus value of Script01.

- *smgt:SensorCollections/1/Sensors/1/SensorID*
  - Points to the SensorID value of the first Sensor Instance of the first SensorCollection Instance.
- *smgt:SensorCollections/#/Sensors/#/SensorID*
  - Points to all SensorID values of the SensorManagement Device.

### A.3.4 Events Properties

Events properties checks if an event has happened. The use into the Condition EBNF indicates that the Condition depends of one of Sensor Management Events (see Section 5.2.7 of SensorManagement Data Model Service []) to be fired.

The Event Property shall use the following template:

- smgt-evt:[event-name]

Where [event-name] is the name of the Event as defined in table 5.3 of SensorManagement Data Model Service.

### A.3.5 Examples

In this section are presented examples of how to use *SensorGroupSets*, *Settings* and *Scripts*. For all examples, the sample *SensorCollection* presented in the following figure and table is used. This sample *SensorCollection* has three (3) light *Sensors* (`LightFloor0001`, `LightFloor0002 and LightCeiling0001`) that are placed in the same living room. Two models are presented. Both models have the same *SensorURN* for the *PowerSwitch DataItem*. However, they differ in the other *SensorURN*, where one model sets only the *ColorRGB DataItem*, and the other the *ColorRGB* and *Brightness DataItems*.

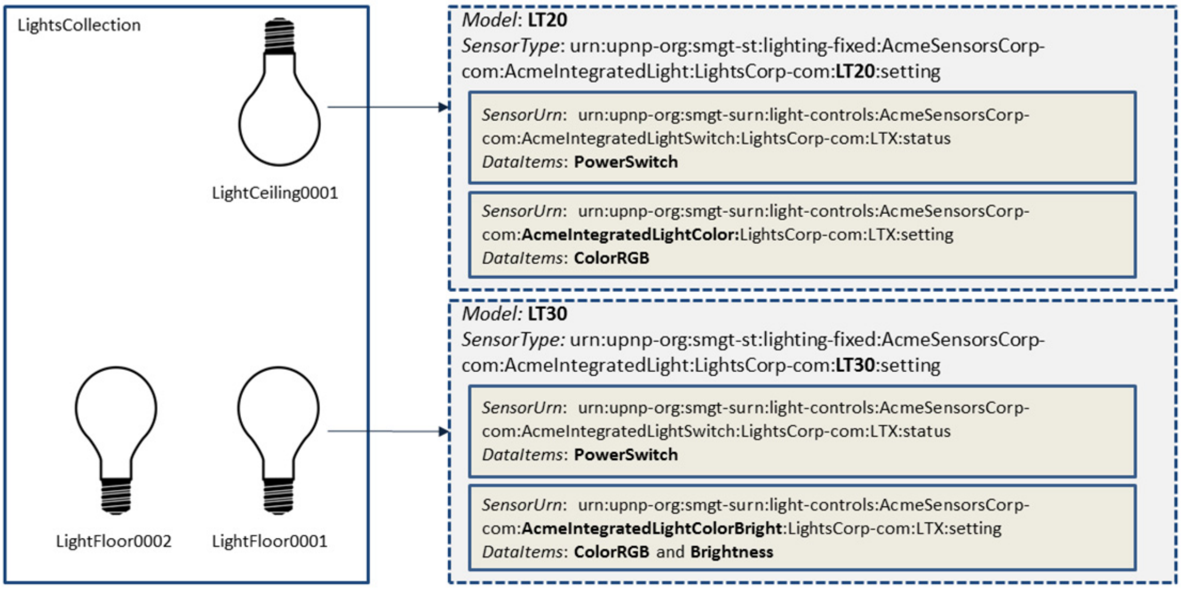
LightsCollection
LightCeiling0001
LightFloor0002
LightFloor0001
*Model*: **LT20**
*SensorType*: urn:upnp-org:smgt-st:lighting-fixed:AcmeSensorsCorp-com:AcmeIntegratedLight:LightsCorp-com:**LT20**:setting
*SensorUrn*: urn:upnp-org:smgt-surn:light-controls:AcmeSensorsCorp-com:AcmeIntegratedLightSwitch:LightsCorp-com:LTX:status
*DataItems*: **PowerSwitch**
*SensorUrn*: urn:upnp-org:smgt-surn:light-controls:AcmeSensorsCorp-com:**AcmeIntegratedLightColor:**LightsCorp-com:LTX:setting
*DataItems*: **ColorRGB**
*Model*: **LT30**
*SensorType*: urn:upnp-org:smgt-st:lighting-fixed:AcmeSensorsCorp-com:AcmeIntegratedLight:LightsCorp-com:**LT30**:setting
*SensorUrn*: urn:upnp-org:smgt-surn:light-controls:AcmeSensorsCorp-com:AcmeIntegratedLightSwitch:LightsCorp-com:LTX:status
*DataItems*: **PowerSwitch**
*SensorUrn*: urn:upnp-org:smgt-surn:light-controls:AcmeSensorsCorp-com:**AcmeIntegratedLightColorBright**:LightsCorp-com:LTX:setting
*DataItems*: **ColorRGB** and **Brightness**

**Figure A.2 — Sample Collection**

**Table A.15 — Sample Collection for Examples**

| Parameters | Value |
|---|---|
| /UPnP/SensorMgt | |
| SensorCollectionsNumberofEntries | 1 |
| SensorCollections/ | |
| 1/CollectionID | `LightsCollection` |
| 1/CollectionType | `urn:upnp-org:smgt-sct:lighting:AcmeSensorsCorp-com:AcmeIntegratedController:LightsRoom:ltroom10` |
| 1/CollectionFriendlyName | `"Living Room Lights"` |
| 1/CollectionInformation | `"Lights Controller Model"` |
| 1/CollectionUniqueIdentifier | `"123456789"` |
| 1/CollectionSpecific | |
| 1/ParticipatingGroupSetID | `LivingRoomCentralLight001` |
| 1/SensorsNumberofEntries | `3` |
| 1/SensorID | `LightCeiling0001` |
| 1/SensorType | `urn:upnp-org:smgt-st:lighting-fixed:AcmeSensorsCorp-com:AcmeIntegratedLight:LightsCorp-com:LT20:setting` |
| 1/SensorUpdateRequest | `0` |
| 1/SensorPollingInterval | `0` |
| 1/SensorReportChangeOnly | `0` |

| | |
|---|---|
| 1/SensorsRelated/ | |
| 1/SensorGroups | |
| 1/SensorDefaultPermissions/ | |
| 1/SensorSpecific | |
| 1/SensorURNsNumberOfEntries | 2 |
| 1/SensorURNs | |
| 1/SensorURN | urn:upnp-org:smgt-surn:light-controls:AcmeSensorsCorp-com:AcmeIntegratedLightSwitch:LightsCorp-com:LTX:status |
| 1/DataItemsNumberOfEntries | 1 |
| 1/DataItems/ | |
| 1/Name | PowerSwitch |
| 1/Type | uda:string |
| 1/Encoding | ascii |
| 1/Description | See Annex E.3.20 |
| 2/SensorURN | urn:upnp-org:smgt-surn:light-controls:AcmeSensorsCorp-com:AcmeIntegratedLightColor:LightsCorp-com:LTX:setting |
| 2/DataItemsNumberOfEntries | 1 |
| 2/DataItems/ | |
| 1/Name | ColorRGB |
| 1/Type | uda:i32 |
| 1/Encoding | ascii |
| 1/Description | See Annex E.3.5 |
| 2/SensorID | LightFloor0001 |
| 2/SensorType | urn:upnp-org:smgt-st:lighting-fixed:AcmeSensorsCorp-com:AcmeIntegratedLight:LightsCorp-com:LT30:setting |
| 2/SensorUpdateRequest | 0 |
| 2/SensorPollingInterval | 0 |
| 2/SensorReportChangeOnly | 0 |
| 2/SensorsRelated/ | |
| 2/SensorGroups | |
| 2/SensorDefaultPermissions/ | |
| 2/SensorSpecific | |
| 2/ParticipatingGroupSetID | LivingRoomFloorLights |
| 2/SensorURNsNumberOfEntries | 2 |
| 2/SensorURNs | |
| 1/SensorURN | urn:upnp-org:smgt-surn:light-controls:AcmeSensorsCorp-com:AcmeIntegratedLightSwitch:LightsCorp-com:LTX:status |
| 1/DataItemsNumberOfEntries | 1 |
| 1/DataItems/ | |
| 1/Name | PowerSwitch |
| 1/Type | uda:string |
| 1/Encoding | ascii |
| 1/Description | See Annex E.3.20 |
| 2/SensorURN | urn:upnp-org:smgt-surn:light-controls:AcmeSensorsCorp-com:AcmeIntegratedLightColorBright:LightsCorp-com:LTX:setting |
| 2/DataItemsNumberOfEntries | 2 |

| | |
|---|---|
| 2/DataItems/ | |
| 1/Name | ColorRGB |
| 1/Type | uda:i32 |
| 1/Encoding | ascii |
| 1/Description | See Annex E.3.5 |
| 2/Name | Brightness |
| 2/Type | uda:i32 |
| 2/Encoding | ascii |
| 2/Description | See Annex E.3.4 |
| 3/SensorID | LightFloor0002 |
| 3/SensorType | urn:upnp-org:smgt-st:lighting-fixed:AcmeSensorsCorp-com:AcmeIntegratedLight:LightsCorp-com:LT30:setting |
| 3/SensorUpdateRequest | 0 |
| 3/SensorPollingInterval | 0 |
| 3/SensorReportChangeOnly | 0 |
| 3/SensorsRelated/ | |
| 3/SensorGroups | |
| 3/SensorDefaultPermissions/ | |
| 3/SensorSpecific | |
| 3/ParticipatingGroupSetID | LivingRoomFloorLights |
| 3/SensorURNsNumberOfEntries | 2 |
| 3/SensorURNs | |
| 1/SensorURN | urn:upnp-org:smgt-surn:light-controls:AcmeSensorsCorp-com:AcmeIntegratedLightSwitch:LightsCorp-com:LTX:status |
| 1/DataItemsNumberOfEntries | 1 |
| 1/DataItems/ | |
| 1/Name | PowerSwitch |
| 1/Type | uda:string |
| 1/Encoding | ascii |
| 1/Description | See Annex E.3.20 |
| 2/SensorURN | urn:upnp-org:smgt-surn:light-controls:AcmeSensorsCorp-com:AcmeIntegratedLightColorBright:LightsCorp-com:LTX:setting |
| 2/DataItemsNumberOfEntries | 2 |
| 2/DataItems/ | |
| 1/Name | ColorRGB |
| 1/Type | uda:i32 |
| 1/Encoding | ascii |
| 1/Description | See Annex E.3.5 |
| 2/Name | Brightness |
| 2/Type | uda:i32 |
| 2/Encoding | ascii |
| 2/Description | See Annex E.3.4 |
| | |
| /UPnP/SensorMgt | |
| SensorGroupSetsNumberofEntries | 2 |

| | |
|---|---|
| SensorGroupSets / | |
| 1/GroupSetID | `LivingRoomCentralLight001` |
| 1/GroupSetType | `smgt-gst:group-control` |
| 1/GroupSetFriendName | `Group of Lights in LivingRoom` |
| 1/GroupSetSpecific/ | |
| 1/MembersNumberOfEntries | `1` |
| 1/Members/ | |
| 1/MemberID | `LightsCollection` |
| 2/GroupSetID | `LivingRoomFloorLights` |
| 2/GroupSetType | `smgt-gst:group-control` |
| 2/GroupSetFriendName | `Group of Lights in LivingRoom Floor for RGB and Brightness control` |
| 2/GroupSetSpecific/ | |
| SensorURNsNumberOfEntries/ | `1` |
| SensorURNs/ | |
| 1/SensorURN | `urn:upnp-org:smgt-surn:light-controls:AcmeSensorsCorp-com:AcmeIntegratedLightColorBright:LightsCorp-com:LTX:setting` |
| 2/Members/ | |
| 1/MemberID | `LightFloor0001` |
| 2/MemberID | `LightFloor0002` |

#### A.3.5.1 Living Room Lights – Simple Example

In this example, the user wants to control all lights in the living room by the use of a *SensorGroupSets*. In the *IoT Management and Control Device* the *SensorGroupSet* `LivingRoomCentralLight001` is defined, where the whole SensorCollection is included, as illustrated in Figure A.3.

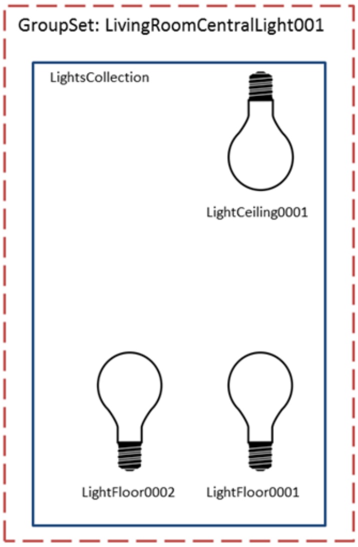

**Figure A.3 — Sample of a GroupSet**

In this usage example, when a *ControlPoint* writes a value to the *PowerSwitch* *DataItem* of `LivingRoomCentralLight001` group, all three children *Sensors* receive the same value. The value is set by the use of *IoT Management and Control TransportGeneric Service* and *WriteSensor* action using the common *SensorURN* `urn:upnp-org:smgt-surn:light-`

controls:AcmeSensorsCorp-com:AcmeIntegratedLightSwitch:LightsCorp-com:LTX:status.

In another usage example, when a *ControlPoint* writes a value to the *Brightness* *DataItem* of LivingRoomFloorLights group, just LightFloor0001 and LightFloor0002 receive this value, as LightCeiling0001 is not included in this group.

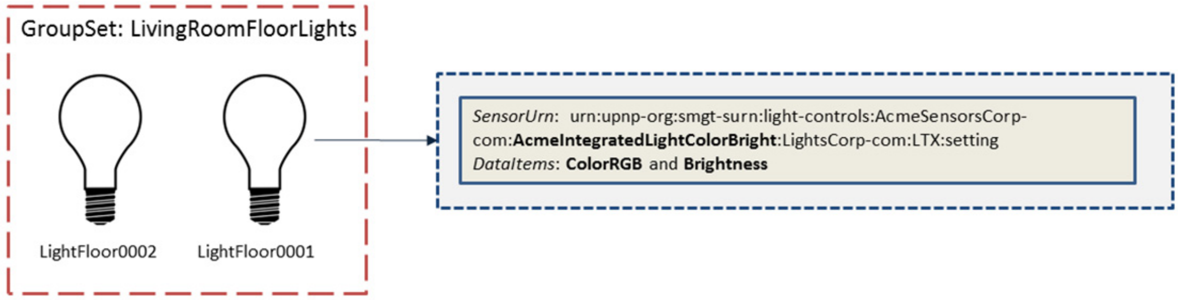
GroupSet: LivingRoomFloorLights
LightFloor0002
LightFloor0001
SensorUrn: urn:upnp-org:smgt-surn:light-controls:AcmeSensorsCorp-com:**AcmeIntegratedLightColorBright**:LightsCorp-com:LTX:setting
DataItems: **ColorRGB** and **Brightness**

**Figure A.4 — Sample of receiving the Brightness setting**

In the latest usage example, if the *ControlPoint* tries to write a value to the *PowerSwitch* *DataItem* of LivingRoomFloorLights groups, a “703 - Sensor URN not found error” is raised, as defined by *IoT Management and Control TransportGeneric Service*.

### A.3.5.2 Living Room Lights – Pre-defined Settings Example

In this example, it is defined a pre-set of values for each light *PowerSwitch* *DataItem* value. In this pre-defined setup, the LightCeiling0001 *PowerSwitch* *DataItem* is set to OFF, and LightFloor0001 and LightFloor0002 *PowerSwitch* *DataItem* is set to ON.
This *Setting* is represented by the Table A.16 with *SettingID* Setting001:

**Table A.16 — Sample Collection for Living Room Lights**

| Parameters | Value |
|---|---|
| /UPnP/ScriptSettings/ | |
| SettingsNumberOfEntries | 1 |
| /UPnP/ScriptSettings/Settings/ | |
| 1/SettingID | Setting001 |
| 1/SettingDescription | "Living Room Lights For Movies" |
| 1/ConfigsNumberOfEntries | 3 |
| 1/Config/ | |
| 1/DeviceID | LightCeiling0001 |
| 1/SensorURN | urn:upnp-org:smgt-surn:light-controls:AcmeSensorsCorp-com:AcmeIntegratedLightSwitch:LightsCorp-com:LTX:status |
| 1/DataRecords | <?xml version="1.0" encoding="UTF-8"?><br><DataRecords xmlns="urn:schemas-upnp-org:ds:drecs"<br>xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"<br>xsi:schemaLocation="urn:schemas-upnp-org:ds:drecs<br>http://www.upnp.org/schemas/ds/drecs-v1.xsd"><br>  <datarecord><br>    <field name="PowerSwitch" type="uda:string" encoding="ascii">**off**</field><br>  </datarecord><br></DataRecords> |

| | |
|---|---|
| 2/DeviceID | LightFloor0001 |
| 2/SensorURN | urn:upnp-org:smgt-surn:light-controls:AcmeSensorsCorp-com:AcmeIntegratedLightSwitch:LightsCorp-com:LTX:status |
| 2/DataRecords | <?xml version="1.0" encoding="UTF-8"?><br><DataRecords xmlns="urn:schemas-upnp-org:ds:drecs"<br>xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"<br>xsi:schemaLocation="urn:schemas-upnp-org:ds:drecs<br>http://www.upnp.org/schemas/ds/drecs-v1.xsd"><br><datarecord><br><field name="PowerSwitch" type="uda:string" encoding="ascii">**on**</field><br></datarecord><br></DataRecords> |
| 3/Device | LightFloor0002 |
| 3/SensorURN | urn:upnp-org:smgt-surn:light-controls:AcmeSensorsCorp-com:AcmeIntegratedLightSwitch:LightsCorp-com:LTX:status |
| 3/DataRecords | <?xml version="1.0" encoding="UTF-8"?><br><DataRecords xmlns="urn:schemas-upnp-org:ds:drecs"<br>xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"<br>xsi:schemaLocation="urn:schemas-upnp-org:ds:drecs<br>http://www.upnp.org/schemas/ds/drecs-v1.xsd"><br><datarecord><br><field name="PowerSwitch" type="uda:string" encoding="ascii">**on**</field><br></datarecord><br></DataRecords> |

As an addition, it is necessary to define one simple *Script* with *ScriptID* *Script001* to activate or deactive *Settings* `Setting001`, as represented in the Table A.17 (which is a continuation of Table A.16):

**Table A.17 — Continue Collection for Living Room Lights**

| Parameters | Value |
|---|---|
| /UPnP/ScriptSettings/ | |
| SettingsNumberOfEntries | 1 |
| … | |
| ScriptsNumberOfEntries | 1 |
| /UPnP/ScriptSettings/Scripts/1/ | |
| 1/ScriptID | Script001 |
| 1/ScriptDescription | Script to apply Settings0001 |
| 1/ScriptAttributes | |
| 1/ScriptStatus | deactivated |
| 1/Condition | true |
| 1/Then | Setting001 |

To apply the `Script001` (and its associated `Setting001`), the *ControlPoint* must write in the paramenter `/UPnP/ScriptSettings/Scripts/1/ScriptStatus` the value of "*activated*", and the *Script* will be executed based on the *Condition*, *Then* and *ScriptAttributes* parameters.

The *Condition* parameter has the value "*true*", which means that the script will be applied every time the *Script* is activated. The *StriptAttributes* parameter is empty, which means that the *Script* does not have any special attribute (such as a timer or period of execution). In the *Then* parameter just the `Setting001` is listed, meaning that only this *Setting* will be applied.

### A.3.5.3 Living Room Lights – Timer Example

In this example it is demonstrated how to apply a *Setting* after a pre-defined timer delay using the *ScriptAttributes* parameter. Table A.18 describes a new *Script* `Script002`.

**Table A.18 — Collection for Living Room Lights - Timer example**

| Parameters | Value |
|---|---|
| /UPnP/ScriptSettings/ | |
| SettingsNumberOfEntries | 1 |
| … | |
| ScriptsNumberOfEntries | `2` |
| /UPnP/ScriptSettings/Scripts/1/ | |
| … | |
| /UPnP/ScriptSettings/Scripts/2/ | |
| 2/ScriptID | `Script002` |
| 2/ScriptDescription | `Script to apply Settings0001 after delay` |
| 2/ScriptAttributes | `delay=00:01:00,delay-expires=00:02:00` |
| 2/ScriptStatus | `deactivated` |
| 2/Condition | `*` |
| 2/Then | `Setting001` |

In this *Script*, when its status is changed to "*activated*", the `Setting001` is applied after the delay of one (1) minute as described on its *ScriptAttributes*. After it is activated, its *ScriptStatus* will be changed to "*deactivated*" two (2) minutes later.

# Annex B Required IoT Management and Control DataItem(s) (normative)

## B.1 Introduction

The following Sensor DataItems shall be supported for all SensorURNs.

**Table B.1 — IoT Management and Control required DataItem(s)**

| Name | Type | Encoding | Description |
|---|---|---|---|
| *ClientID* | *xsd:string* | *string* | The required *ClientID* DataItem shall match the value of the *SensorClientID* argument of the SensorTransportGeneric service *ConnectSensor()* or *ReadSensor()* actions.<br>This DataItem can be used by sensor transport clients to label the transport connection (or SOAP action) which is requesting the sensor data. |
| *ReceiveTimestamp* | *xsd:dateTime* | *string* | The required *ReceiveTimestamp* DataItem shall provide a date-time stamp indicating when the sensor data was received by the IoT Management and Control device<br>For example both of these timestamps represent the same time:<br>2013-02-01T20:00:01Z<br>2013-02-01T12:00:00-08:00 |

# Annex C Common Device Identifiers (normative)

## C.1 Introduction

The following table lists common device identifiers. If one of these device identifiers apply to the IoT Management and Control application, then this value should be used in URN [device-identifier] fields.

**Table C.1 — Common Device Identifiers**

| id | Category and Device | Comment | subid | Subclassifcation (incl. UPnP classification) |
|---|---|---|---|---|
| | **Space Conditioning (11)** | | | |
| 1 | Unitary System | window and wall units without external components; self-contained | 1S1 | *Climate* |
| 2 | Boiler | device that heats water (or other liquid). | 2S1 | *HotWater_Heater* |
| 3 | Furnace | device that heats air (or other gas) | | |
| 4 | Pump | device that moves water (or other liquid) | | |
| 5 | Fan | device that moves air (or other gas) | | |
| 6 | Condensing Unit | always includes a compressor | | |
| 7 | Condensor | no compressor; just fan | | |
| 8 | Humidifier | adds moisture to air | | |
| 9 | Dehumidifier | removes moisture from air | | |
| 10 | HVAC - control | directs operation of other HVAC devices; includes HVAC sensors | 10S1 | *Controller* |
| | | | 10S2 | *HVAC* |
| | | | 10S3 | *Thermostat* |
| | | | 10S4 | Damper |
| 11 | HVAC - other | not readily classified into any of the above | 11S1 | *HVAC* |
| | | | 11S2 | *RoomAirConditioner* |
| | | | | |
| | **Lighting (5)** | | | |
| 12 | Lighting - outdoor | | | |
| 13 | Lighting - fixed | hardwired | 13S1 | *Light* |
| 14 | Lighting - portable | has normal plug | | |
| 15 | Lighting - controls | incl. sensors for lighting | | |
| 16 | Lighting - other | not readily classified into any of the above | | |
| | | | | |
| | **Appliance (10)** | large devices; small in misc. | | |

| | | | | |
|---|---|---|---|---|
| 17 | Clothes dryer | | 17S1 | *Dryer* |
| 18 | Clothes washer | incl. combined washer/dryer | 18S1 | *Washing machine* |
| 19 | Dishwasher | | | |
| 20 | Freezer | | | |
| 21 | Ice machine | | | |
| 22 | Oven | incl. warming drawer | | |
| 23 | Range | incl. cooktop and combined cooktop/oven | | |
| 24 | Refrigerator | including wine coolers, fridge/freezer combo | | |
| 25 | Water heater | | | |
| 26 | Appliance - other | not readily classified into any of the above | 26S1 | *Microwave* |
| | | | 26S2 | Coffee Maker |
| | | | | |
| | **Electronics (21)** | | | |
| 27 | Audio system | integrated source and speaker; incl. radio, boombox | | |
| 28 | Audio/video player | e.g. CD, DVD, VCR, cassette, turntable | 28S1 | *MediaRenderer* |
| 29 | Camera | | | |
| 30 | Computer, desktop | incl. integrated | | |
| 31 | Computer, notebook | | | |
| 32 | Computer, server | | | |
| 33 | Computer, other | | | |
| 34 | Data storage | | | |
| 35 | Display | incl. monitor, projectors, TVs, digital picture frame | | |
| 36 | Electronics - portable | can be operated by battery and not otherwise classified | | |
| 37 | Game console | | | |
| 38 | Imaging equipment | fax, multi-function device, scanner, printer, label printer | | |
| 39 | Musical instrument | also incl. recording devices, mixers, amplifiers | | |
| 40 | Network equipment | modems, switches, routers, access points, etc. | | |
| 41 | Phone handset | incl. tablet | | |
| 42 | Receiver | incl. amplifier, home theatre system | 42S1 | Audio Amp |
| | | | 42S2 | Tuner |
| | | | 42S3 | Audio Equalizer |
| | | | 42S4 | A/V Switch |

| | | | | |
|---|---|---|---|---|
| 43 | Set-top box | cable, satellite, Internet | | |
| 44 | Telephony | base stations, answering machines, corded phones | | |
| 45 | Television | | | |
| 46 | Audio/video - other | | | |
| 47 | Electronics - other | not readily classified into any of the above, incl. clocks | 47S1 | MediaServer |
| | | | 47S2 | Intercom |
| | | | | |
| | **Miscellaneous (29)** | | | |
| 48 | Agriculture | incl. irrigation timers | 48S1 | LawnSprinkler |
| 49 | Air compressors | | | |
| 50 | Bathroom device | toilets, faucets, hand dryers, etc. | | |
| 51 | Battery charger | | | |
| 52 | Business equipment | money or office paper related | | |
| 53 | Cleaning equipment | incl. vacuum | | |
| 54 | Cooking - portable | | | |
| 55 | Decorations/hobby/leisure | fountains, aquaria, kilns, baby rockers, massage chairs | 55S1 | Fountain |
| 56 | Entertainment | gambling, arcade, etc. | | |
| 57 | Exercise machine | | | |
| 58 | Food service/prep | ice cream machine, fryer, steam cooker | | |
| 59 | Household | non-food; incld iron, sewing machine, … | | |
| 60 | HVAC - portable | humidifiers, portable fans, heaters | | |
| 61 | Industrial | process equipment and related | | |
| 62 | Kitchen | small devices | | |
| 63 | Laboratory equipment | | | |
| 64 | Lighting - decorative | | 64S1 | Light |
| 65 | Lighting - emergency | incl. exit lights | | |
| 66 | Medical equipment | for professional settings | | |
| 67 | Microwave oven | | | |
| 68 | Outdoor appliance | mowers, trimmers, snow melting coils, grill | | |
| 69 | Personal - health | hair dryers, curlers, shavers, trimmers, toothbrush | | |
| 70 | Pool - spa | | | |
| 71 | Sanitation | garbage disposal, trash | 71S1 | Disposal |

| | | | | |
|---|---|---|---|---|
| | | compactor, wastewater pump, water filter | 71S2 | TrashCompactor |
| 72 | Sign | not incl. electronic displays | | |
| 73 | Tool - construction | | | |
| 74 | Tool - non-construction | e.g. auto, beauty | | |
| 75 | Vending machine | | | |
| 76 | Water dispenser | | | |
| | | | | |
| | **Infrastructure (11)** | Devices not used directly by people | | |
| 77 | Breakers | incl. AFI/GFCI | 77S1 | Load Center |
| 78 | Doors / Windows | incl. garage doors, gates, window shades | 78S1 | Door_Lock |
| | | | 78S2 | Door_Sensor |
| | | | 78S3 | Window_Sensor |
| | | | 78S4 | Keypad |
| | | | 78S5 | Keyfob |
| | | | 78S6 | GarageDoorOpener |
| | | | 78S7 | WindowCovering |
| | | | 78S8 | Skylight |
| 79 | Fireplace | | | |
| 80 | Motor | actuators | 80S1 | Actuator |
| 81 | Power - portable | power strips, surge protectors, UPS, timer | | |
| 82 | Power - fixed | transformers, switchgears, inverters, voltage regulators, power conditioners | | |
| 83 | Pump | | 83S1 | HeatPump |
| | | | 83S2 | PoolPump |
| 84 | Security | security cameras, systems | 84S1 | Surveillance |
| | | | 84S2 | Alarm_Panel |
| | | | 84S3 | Panic_Pendant |
| 85 | Sensors | incl. detectors: fire, smoke, gas, fluids | 85S1 | Sensor |
| | | | 85S2 | Motion_Sensor |
| | | | 85S3 | Occupancy_Sensor |
| | | | 85S4 | Temperature_Sensor |
| | | | 85S5 | Glass_break_Sensor |
| | | | 85S6 | CO_Sensor |
| | | | 85S7 | Heat_Sensor |
| | | | 85S8 | Smoke_Sensor |
| | | | 85S9 | Flammable_Gas_Sensor |
| | | | 85S10 | Humidity_Sensor |
| | | | 85S11 | Wind_Speed_Sensor |
| | | | 85S12 | Wind_Direction_Sensor |
| | | | 85S13 | Rain_Sensor |

| | | | | |
|---|---|---|---|---|
| | | | 85S14 | Dewpoint_Sensor |
| | | | 85S15 | Barometric_Pressure_Sensor |
| | | | 85S16 | Stress_Sensor |
| | | | 85S17 | Flux_Sensor |
| | | | 85S18 | Weight_Sensor |
| | | | 85S19 | Light Sensor |
| 86 | Signage | | | |
| 87 | Meter | utility and other | 87S1 | Power meter |
| | | | | |
| | **Transportation (4)** | | | |
| 88 | Transport, fixed | elevator, escalator, lifts, etc. | | |
| 89 | Vehicle-large | | | |
| 90 | Vehicle-small | incl. wheelchair, golf cart | | |
| 91 | Transport, other | incl. auto engine heater, vehicle charger, parking meter, parking equipment | | |
| | | | | |
| | **Other (1)** | | | |
| 92 | Other | truly unclassifiable | 92S1 | Safety |

Table C.1 was prepared at Lawrence Berkeley National Laboratory under Contract No. DE-AC02-05CH11231 with the U.S. Department of Energy."

**Table C.2 — mapping of mandatory DataItems per Common Device Identifier**

| id | Category and Device | Related Mandatory DataItem (unless specified optional) | Section |
|---|---|---|---|
| | **Space Conditioning (11)** | | |
| 1 | Unitary System | | |
| 1S1 | Climate | Switch | E.3.30 |
| | | Temperature | E.3.31 |
| 2 | Boiler | Switch | E.3.30 |
| | | Temperature | E.3.31 |
| 2S1 | HotWater_Heater | Switch | E.3.30 |
| | | Temperature | E.3.31 |
| 3 | Furnace | Switch | E.3.30 |
| | | Temperature | E.3.31 |
| 4 | Pump | Switch | E.3.30 |
| | | Flow | E.3.10 |
| 5 | Fan | Switch | E.3.30 |
| | | Direction (optional) | E.3.9 |
| | | RelSpeed (optional) | E.3.26 |
| 6 | Condensing Unit | Switch | E.3.30 |
| 7 | Condensor | Switch | E.3.30 |

| | | | |
|---|---|---|---|
| 8 | Humidifier | *Switch* | E.3.30 |
| | | *Humidity* | E.3.14 |
| 9 | Dehumidifier | *Switch* | E.3.30 |
| | | *Humidity* | E.3.14 |
| 10 | HVAC - control | | |
| 10S1 | *Controller* | | |
| 10S2 | *HVAC* | | |
| 10S3 | *Thermostat* | *Switch* | E.3.30 |
| | | *Temperature* | E.3.31 |
| | | *Humidity* (optional) | E.3.14 |
| 10S4 | Damper | *Switch* | E.3.30 |
| 11 | HVAC - other | | |
| 11S1 | *HVAC* | | |
| 11S2 | *RoomAirConditioner* | *Switch* | E.3.30 |
| | | *Temperature* | E.3.31 |
| | | *Humidity* (*optional*) | E.3.14 |
| | | | |
| | **Lighting (5)** | | |
| 12 | Lighting - outdoor | *Switch* | E.3.30 |
| 13 | Lighting - fixed | *Switch* | E.3.30 |
| 13S1 | *Light* | *Switch* | E.3.30 |
| | | *Brightness* (*optional*) | E.3.4 |
| | | *ColorRGB* (*optional*) | E.3.5 |
| 14 | Lighting - portable | *Switch* | E.3.30 |
| 15 | Lighting - controls | *Switch* | E.3.30 |
| 16 | Lighting - other | *Switch* | E.3.30 |
| | | | |
| | **Appliance (10)** | | |
| 17 | Clothes dryer | *Switch* | E.3.30 |
| | | *DryingState* | E.3.40 |
| 17S1 | *Dryer* | *Switch* | E.3.30 |
| | | *DryingState* | E.3.40 |
| 18 | Clothes washer | *Switch* | E.3.30 |
| | | *WashState* | E.3.45 |
| 18S1 | *Washing machine* | *Switch* | E.3.30 |
| | | *WashState* | E.3.45 |
| 19 | Dishwasher | *Switch* | E.3.30 |
| | | *WashState* | E.3.45 |
| | | *DryingState* | E.3.40 |
| 20 | Freezer | *Switch* | E.3.30 |
| | | *Temperature* | E.3.31 |
| 21 | Ice machine | *Switch* | E.3.30 |
| | | *Temperature* | E.3.31 |
| 22 | Oven | *Switch* | E.3.30 |
| | | *Temperature* | E.3.31 |
| 23 | Range | *Switch* | E.3.30 |

| | | | |
|---|---|---|---|
| | | Temperature | E.3.31 |
| 24 | Refrigerator | Switch | E.3.30 |
| | | Temperature | E.3.31 |
| 25 | Water heater | Switch | E.3.30 |
| | | Temperature | E.3.31 |
| 26 | Appliance - other | Switch | E.3.30 |
| 26S1 | Microwave | Switch | E.3.30 |
| | | Power | E.3.19 |
| 26S2 | Coffee Maker | Switch | E.3.30 |
| | | | |
| | **Electronics (21)** | | |
| 27 | Audio system | Switch | E.3.30 |
| 28 | Audio/video player | Switch | E.3.30 |
| 28S1 | MediaRenderer | Switch | E.3.30 |
| 29 | Camera | Switch | E.3.30 |
| 30 | Computer, desktop | Switch | E.3.30 |
| 31 | Computer, notebook | Switch | E.3.30 |
| 32 | Computer, server | Switch | E.3.30 |
| 33 | Computer, other | Switch | E.3.30 |
| 34 | Data storage | Switch | E.3.30 |
| 35 | Display | Switch | E.3.30 |
| 36 | Electronics - portable | BatteryState | E.3.37 |
| 37 | Game console | Switch | E.3.30 |
| 38 | Imaging equipment | Switch | E.3.30 |
| 39 | Musical instrument | Switch | E.3.30 |
| 40 | Network equipment | Switch | E.3.30 |
| 41 | Phone handset | BatteryState | E.3.37 |
| 42 | Receiver | Switch | E.3.30 |
| 42S1 | Audio Amp | Switch | E.3.30 |
| 42S2 | Tuner | Switch | E.3.30 |
| 42S3 | Audio Equalizer | Switch | E.3.30 |
| 42S4 | A/V Switch | Switch | E.3.30 |
| 43 | Set-top box | Switch | E.3.30 |
| 44 | Telephony | Switch | E.3.30 |
| 45 | Television | Switch | E.3.30 |
| 46 | Audio/video - other | Switch | E.3.30 |
| 47 | Electronics - other | Switch | E.3.30 |
| 47S1 | MediaServer | Switch | E.3.30 |
| 47S2 | Intercom | Switch | E.3.30 |
| | | | |
| | **Miscellaneous (29)** | | |
| 48 | Agriculture | | |
| 48S1 | LawnSprinkler | Switch | E.3.30 |
| 49 | Air compressors | Switch | E.3.30 |
| 50 | Bathroom device | Switch | E.3.30 |
| 51 | Battery charger | Switch | E.3.30 |

| | | | |
|---|---|---|---|
| | | Batterystate | E.3.37 |
| 52 | Business equipment | | |
| 53 | Cleaning equipment | Switch | E.3.30 |
| | | CleanerState | E.3.38 |
| 54 | Cooking - portable | | |
| 55 | Decorations/hobby/leisure | | |
| 55S1 | Fountain | Switch | E.3.30 |
| 56 | Entertainment | | |
| 57 | Exercise machine | Switch | E.3.30 |
| 58 | Food service/prep | Switch | E.3.30 |
| 59 | Household | Switch | E.3.30 |
| 60 | HVAC - portable | | |
| 61 | Industrial | | |
| 62 | Kitchen | | |
| 63 | Laboratory equipment | | |
| 64 | Lighting - decorative | Switch | E.3.30 |
| 64S1 | Light | Switch | E.3.30 |
| 65 | Lighting - emergency | Switch | E.3.30 |
| 66 | Medical equipment | | |
| 67 | Microwave oven | Switch | E.3.30 |
| 68 | Outdoor appliance | | |
| 69 | Personal - health | | |
| 70 | Pool - spa | | |
| 71 | Sanitation | | |
| 71S1 | Disposal | Switch | E.3.30 |
| 71S2 | TrashCompactor | Switch | E.3.30 |
| 72 | Sign | | |
| 73 | Tool - construction | | |
| 74 | Tool - non-construction | | |
| 75 | Vending machine | Switch | E.3.30 |
| 76 | Water dispenser | Switch | E.3.30 |
| | | | |
| | **Infrastructure (11)** | | |
| 77 | Breakers | | |
| 77S1 | Load Center | EnergyOverloadDetector | E.3.47 |
| 78 | Doors / Windows | Lock | E.3.42 |
| 78S1 | Door_Lock | Lock | E.3.42 |
| 78S2 | Door_Sensor | RelPosition | E.3.24 |
| 78S3 | Window_Sensor | RelPosition | E.3.24 |
| 78S4 | Keypad | | |
| 78S5 | Keyfob | | |
| 78S6 | GarageDoorOpener | Lock | E.3.42 |
| | | RelPosition | E.3.24 |
| 78S7 | WindowCovering | RelPosition | E.3.24 |
| 78S8 | Skylight | Lock | E.3.42 |
| | | RelPosition | E.3.24 |

| | | | |
|---|---|---|---|
| 79 | Fireplace | *Switch* | E.3.30 |
| 80 | Motor | *Switch* | E.3.30 |
| | | *RelPosition (optional)* | E.3.24 |
| 80S1 | *Actuator* | *Switch* | E.3.30 |
| | | *RelPosition (optional)* | E.3.24 |
| 81 | Power - portable | | |
| 82 | Power - fixed | | |
| 83 | Pump | *Switch* | E.3.30 |
| 83S1 | *HeatPump* | *Switch* | E.3.30 |
| 83S2 | *PoolPump* | *Switch* | E.3.30 |
| 84 | Security | | |
| 84S1 | *Surveillance* | *Switch* | E.3.30 |
| 84S2 | *Alarm_Panel* | | |
| 84S3 | *Panic_Pendant* | | |
| 85 | Sensors | | |
| 85S1 | *Sensor* | *Count* | E.3.7 |
| 85S2 | *Motion_Sensor* | *MovementDetector* | E.3.47 |
| 85S3 | *Occupancy_Sensor* | *OccupanyDetector* | E.3.47 |
| 85S4 | *Temperature_Sensor* | *HeatDetector* | E.3.47 |
| | | *Temperature* | E.3.31 |
| 85S5 | *Glass_break_Sensor* | *GlassBreakDetector* | E.3.47 |
| 85S6 | *CO_Sensor* | *CODetector* | E.3.47 |
| 85S7 | *Heat_Sensor* | *HeatDetector* | E.3.47 |
| 85S8 | *Smoke_Sensor* | *SmokeDetector* | E.3.47 |
| 85S9 | *Flammable_Gas_Sensor* | *GasDetector* | E.3.47 |
| 85S10 | *Humidity_Sensor* | *Humidity* | E.3.14 |
| 85S11 | *Wind_Speed_Sensor* | *Speed* | E.3.28 |
| 85S12 | *Wind_Direction_Sensor* | *WRDirection* | E.3.36 |
| 85S13 | *Rain_Sensor* | *RainDetector* | E.3.47 |
| 85S14 | *Dewpoint_Sensor* | *Dewpoint* | E.3.48 |
| 85S15 | *Barometric_Pressure_Sensor* | *Pressure* | E.3.22 |
| 85S16 | *Stress_Sensor* | *Force* | E.3.50 |
| 85S17 | *Flux_Sensor* | *Flow* | E.3.10 |
| 85S18 | *Weight_Sensor* | *Mass* | E.3.15 |
| 85S19 | *Light Sensor* | *LightDetector* | E.3.47 |
| 86 | Signage | | |
| 87 | Meter | *Count* | E.3.7 |
| 87S1 | Power meter | *Power* | E.3.19 |
| | | | |
| | **Transportation (4)** | | |
| 88 | Transport, fixed | | |
| 89 | Vehicle-large | | |
| 90 | Vehicle-small | | |
| 91 | Transport, other | | |
| | | | |

| | | | |
|---|---|---|---|
| | **Other (1)** | | |
| 92 | Other | | |
| 92S1 | *Safety* | | |

# Annex D IEEE-11073 Personal Health Devices

## D.1 IoT Mangement and Control for IEEE-11073 Personal Health Medical Device Systems

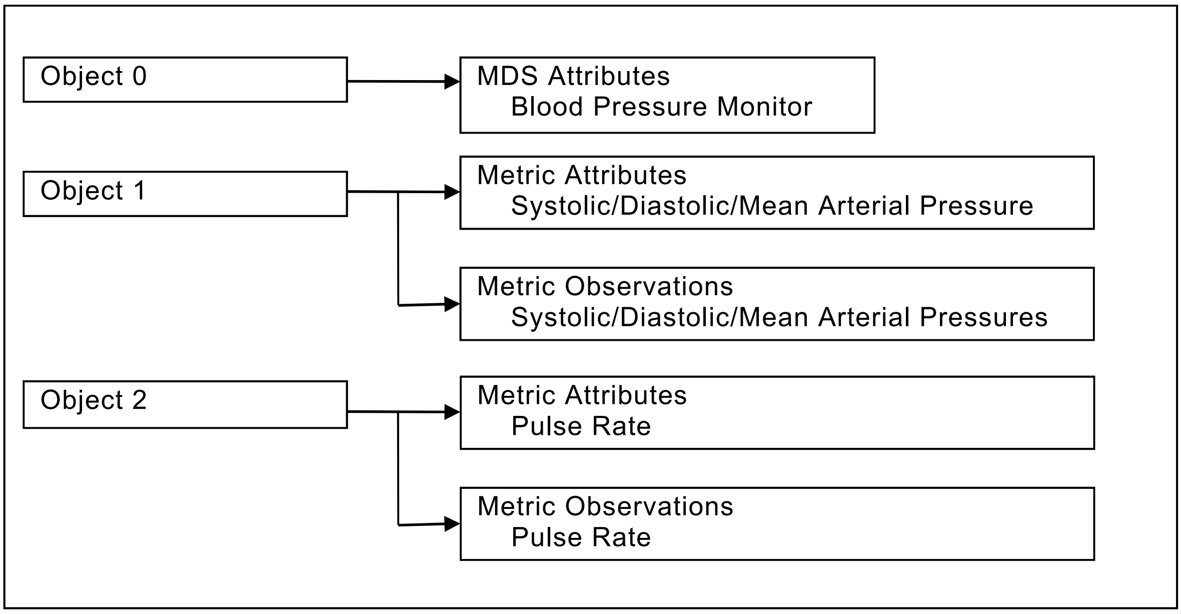

**Figure D.1 —** Blood Pressure Monitor – Medical Device System (Informative)

This annex describes UPnP *IOT Management and Control* device requirements for support of IEEE-11073 Personal Health Medical Device Systems (agents). A UPnP IoT Management and Control device supporting IEEE-11073 Personal Health Devices encapsulates the functionality provided by an IEEE-11073 Manager which in turn implements IEEE-11073 communication protocols per IEEE-11073 Part 20601: Application Profile - Optimized Exchange [8]. IEEE-11073 Medical Device Systems are described as a set of objects. At a top level, Object 0 represents the properties of the medical device itself including identifying information for the device. Objects 1-n describe various measurement functions the medical device can perform. IEEE-11073 protocols separate data description and data transmission, therefore received measurement data (observations) only contain the minimal data relating to the quantities being measured. The attributes of the corresponding measurement object provide context such as measurement type, units of measure and observation data formats. Depending on the medical device being described measurement data may contain both time information as well as multiple measured quantities. Therefore, it is important that both measurement data and measurement object attributes are provided to UPnP clients processing data from IEEE-11073 agent devices. See the IEEE-11073-104zz Device specialization [9] series for detailed information on specific measurement devices. Attribute information from medical devices is conveyed in a tag-length-value format based on IEEE-11073 Medical Device Encoding Rules (MDER) as described in IEEE-11073 Part 20601: Application Profile - Optimized Exchange.

UPnP IoT Management and Control exposes IEEE-11073 Medical Devices to UPnP clients as a sensor collection and sensor. The sensor collection exposes identifying information extracted from the Object 0 of the Medical Device System. The Sensor supports a set of DataItem(s) representing the attributes and observation data for each medical device object reported by the IEEE-11073 Manager component. Additional DataItem(s) supply timestamps for the time of observation (if available) as well as the time of reception by the UPnP IoT Management and Control device.

UPnP IoT Management and Control support of Persistent Metric Storage objects (PM-store) is described in Annex D.4.

### D.1.1 IEEE-11073 Personal Health Device Parameters

This section describes extensions to the IoT Management and Control General Data Model for IEEE-11073 Personal Health Devices.

**Table D.1 — IEEE-11073 specific Medical Device System Parameters**

| Name | Type | Acc | Req | Reference | EOC | Ver |
|---|---|---|---|---|---|---|
| /UPnP/SensorMgt/SensorCollections/#/CollectionSpecific/IEEE-11073/ | SingleInstance | | | D.1.1.2 | | |
| SystemType | string | | | D.1.1.3 | | |
| SystemModel | string | | | D.1.1.4 | | |
| SystemID | string | | | D.1.1.5 | | |
| ConfigurationsNumberOfEntries | int | | | | | |
| | | | | | | |
| /UPnP/SensorMgt/SensorCollections/#/CollectionSpecific/IEEE11073/Association/ | SingleInstance | | | D.1.1.6 | | |
| RequestConfig | int | | | D.1.1.7 | | |
| CurrentConfig | int | | | D.1.1.8 | | |
| LastConfig | int | | | D.1.1.9 | | |
| ConfigurationsNumberOfEntries | int | | | | | |
| | | | | | | |
| /UPnP/SensorMgt/SensorCollections/#/CollectionSpecific/IEEE11073/Configurations/#/ | MultiInstance | | | | | |
| ConfigID | int | | | | | |
| ConfigData | string | | | | | |
| | | | | | | |

#### D.1.1.1 /UPnP/SensorMgt/SensorCollections/#/CollectionType

**Type:** string

**Description:** The following *CollectionType* value shall be used for IEEE-11073 Personal Health Devices.

```
MDS-Collection-Type ::=  urn:upnp-org:smgt-sct:ieee-11073_mds:ieee-org:11073_104zzz
                         :<MDS-vendor>:<MDS-model>
```

Where: 11073_104zzz indicates the IEEE document number defining the Medical Device Specialization.

#### D.1.1.2 /UPnP/SensorMgt/SensorCollections/#/CollectionSpecific/IEEE-11073/

**Type:** SingleInstance

**Description:** The conditionally required *IEEE-11073 SingleInstance* node shall be included for IEEE-11073 Personal Health Devices.

#### D.1.1.3 /UPnP/SensorMgt/SensorCollections/#/CollectionSpecific/IEEE-11073/SystemType

**Type:** CSV(string)

**Description:** The required *SystemType parameter* indicates the type of IEEE-11073 Medical Device System. The contents of the *SystemType parameter* correspond to the *System-Type* or *System-Type-Spec-List* attribute(s) as described in IEEE-11073 Part 20601: Application Profile - Optimized Exchange [8] - MDS Class.Values for the *SystemType parameter* shall be presented in text format as defined by the individual IEEE-11073-104zz Device specialization [9] documents.

For example a IEEE-11073 Blood Pressure measurement device is encoded as the CSV(string) value:

```
MDC_DEV_SPEC_PROFILE_BP, 1
```

#### D.1.1.4 /UPnP/SensorMgt/SensorCollections/#/CollectionSpecific/IEEE-11073/SystemModel

**Type:** CSV(string)
**Description:** The required *SystemModel* parameter indicates the manufacturer and model of a IEEE-11073 Medical Device System. The contents of the *SystemModel* parameter correspond to the *System-Model* attribute as described in IEEE-11073 Part 20601: Application Profile - Optimized Exchange [8] - MDS Class Values for the *SystemModel* parameter shall be presented in text format as defined by individual IEEE-11074-104zz Device specialization [9] documents.

For example, a IEEE-11073 Blood Pressure measurement device with Device-Vendor: `The Company` and with Device-Model: `TheBPM ABC`, is encoded as the CSV(**string**) value consisting of the Device-Vendor identification followed by the Device-Model identification:

```
TheComany,TheBPM ABC
```

#### D.1.1.5 /UPnP/SensorMgt/SensorCollections/#/CollectionSpecific/IEEE-11073/SystemID

**Type:** string
**Description:** The required *SystemID parameter* provides a 64-bit extended unique identifier value for the IEEE-11073 Medical Device System. The contents of the *SystemID parameter* correspond to the *System-Id* attribute as described in IEEE-11073 Part 20601: Application Profile - Optimized Exchange [8] - MDS Class. The values for this *parameter* shall be presented as a 16-digit character hexadecimal string.

For example a IEEE-11073 Blood Pressure measurement device would be encoded as:

*1122334455667707*

#### D.1.1.6 /UPnP/SensorMgt/SensorCollections/#/CollectionSpecific/IEEE-11073/Association

**Type:** SingleInstance
**Description:** The required *Association SingleInstance* node includes parameters which control the association process for IEEE-11073 Medical Device Systems.

#### D.1.1.7 /UPnP/SensorMgt/SensorCollections/#/CollectionSpecific/IEEE-11073/Association/RequestConfig

**Type:** int
**Description:** The required *RequestConfig parameter* shall request the IoT Management and Control device begin a new association process with the corresponding Medical Device System. This value corresponds to the *Dev-Config-Id* value as described in IEEE-11073 Part 20601: Application Profile - Optimized Exchange [8]**.** If this parameter is written to ("0")**,** the IoT Management and Control device shall terminate any current association and shall disable further associations with this IoT Management and Control device. If this parameter is set to ("-1"), then the IoT Management and Control device shall automatically associate with any acceptable configuration presented by the corresponding IEEE-11073 Medical Device System.

#### D.1.1.8 /UPnP/SensorMgt/SensorCollections/#/CollectionSpecific/IEEE-11073/Association/CurrentConfig

**Type:** int
**Description:** The required *CurrentConfig parameter* shall indicate the current association for the corresponding Medical Device System. If there is no current association, then this parameter shall be ("0").

### D.1.1.9 /UPnP/SensorMgt/SensorCollections/#/CollectionSpecific/IEEE-11073/Association/LastConfig

**Type:** int

**Description:** The required *LastConfig* *parameter* shall indicate the last (or currently active) associated configuration for the corresponding Medical Device System. This parameter shall retain its value regardless of whether or not the Medical Device System is currently associated. If there was no known prior associated configuration this parameter shall be ("0").

## D.2 IoT Management and Control for IEEE-11073 Medical Device Sensors

## D.3 IoT Management and Control Parameters for IEEE-11073 Personal Health Sensors

The UPnP IoT Management and Control device represents IEEE-11073 Personal Health Devices as a SensorCollection and an included Sensor. The Sensor shall include the following parameters:

**Table D.2 — IEEE-11073 Medical Device Sensor Parameters**

| Name | Type | Acc | Req | Reference | EOC | Ver |
|---|---|---|---|---|---|---|
| /UPnP/SensorMgt/SensorCollections/#/Sensors/# | MultiInstance | | | | | |
| SensorType | string | | | D.3.1.1 | | |
| | | | | | | |
| /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorURNs/# | MultiInstance | | | | | |
| SensorURN | string | | | D.3.1.3 | | |
| | | | | | | |
| /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorURNs/#/DataItems/# | MultiInstance | | | | | |
| Name | string | | | D.3.1.5,<br>D.3.1.6 | | |
| Type | string | | | D.3.1.5,<br>D.3.1.6 | | |
| Encoding | string | | | D.3.1.5,<br>D.3.1.6 | | |

### D.3.1.1 /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorType

**Type:** string

**Description:** The following *SensorType* value shall be used for IEEE-11073 Medical Device Systems:

```
MOC-Sensor-Type ::=   urn:upnp-org:smgt-st:ieee-11073_mds:ieee-org:11073_104zzz
                      :<MDS-vendor>:<MDS-model>:telemetry
```

Where: 11073_104zzz indicates the IEEE document number defining the Medical Device Specialization.

### D.3.1.2 /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorSpecific/

**Type:** SingleInstance

**Description:** There are no defined child nodes for IEEE-11073 Medical Object Class sensors.

### D.3.1.3 /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorURNs/#/SensorURN

**Type:** string

**Description:** The *SensorURN parameter* for IEEE-11073 Medical Object Class Sensors shall use the following value:

```
MOC-SensorURN ::=  urn:upnp-org:smgt-surn:ieee-11073_mds:ieee-org:11073_104zzz
                   :<MDS-vendor>:<MDS-model>:telemetry
```

Where: 11073_104zzz indicates the IEEE document number defining the Medical Device Specialization.

### D.3.1.4 DataItems for IEEE-11073 Medical Object Class Sensors

**Type:** MultiInstance

**Description:** The IEEE-11073 SensorURN shall support the following DataItem(s) as listed in the table below. Each DataItem shall be described by a set of three parameter nodes:

*/UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorURNs/#/SensorURN/DataItems/#/*

*Name*

*Type*

*Encoding*

The following table lists the settings for the three parameters above to generate the named DataItem described.

Note: These Sensor parameters do not directly report sensor data, instead they describe named DataItems which are returned by the sensor using either SOAP actions or transport connections to the sensor.

**Table D.3 — Default DataItem(s) for IEEE-11073 Medical Object Class Sensors**

| Name | Type | Encoding | Description |
|---|---|---|---|
| *ClientID* | *xsd:string* | *string* | *ClientID* argument value provided from the SensorTransportGeneric ConnectSensor() action. |
| *ReceiveTimeStamp* | *xsd:dateTime* | *string* | The required *ReceiveTimestamp* DataItem for an IEEE-11073 Sensor shall provide a date-time stamp indicating when the Medical Device observation was received by the IoT Management and Control device.<br>For example:<br>`2013-02-01T20:00:01Z`<br>`2013-02-01T12:00:00-08:00` |
| *ObservationTimeStamp* | *xsd:dateTime* | *string* | The required *ObservationTimestamp* DataItem for an IEEE-11073 Sensor shall provide a date-time stamp indicating when the observation was recorded by the Medical Device. If the Medical Device observation does not provide timestamp information, the value of this DataItem shall be empty.<br>For example:<br>`2013-02-01T20:00:01Z`<br>`2013-02-01T12:00:00-08:00` |
| *PersonID* | *int* | *string* | The allowed PersonID corresponds to the IEEE-11073 *person-id* attribute. |

### D.3.1.5 DataItem(s) for IEEE-11073 Medical Object Class Sensor Attributes

**Type:** MultiInstance
**Description:** The IEEE-11073 SensorURN shall support the following DataItem(s) to report the attributes of each Medical Object Class Handle within a IEEE-11073 Medical Device System. Each DataItem shall be described by a set of three parameter nodes:

*/UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorURNs/#/SensorURN/DataItems/#/*

*Name*

*Type*

*Encoding*

The following table summarizes the settings for the three parameter values listed above which define named DataItem(s) that return attributes for IEEE-11073 Medical Objects.

Note: These Sensor parameters do not directly report sensor data, instead they describe named DataItem(s) which are returned by the sensor using either SOAP actions or transport connections to the sensor. See Sensor Transport Generic specification [12] for further details.

**Table D.4 — DataItem(s) for Medical Object Class Attributes**

| Name | Type | Encoding |
|---|---|---|
| $[MDS-Handle]-Attr | mds:MDSAttributes | Base64 |
| $[MDS-Handle]-Attr | mds:MOCMetricAttributes | Base64 |

Where: *[MDS-Handle]* is substituted with the Handle value of the corresponding Medical Device System Object.

For MDS Object 0, the DataItem *Type* parameter value `IEEE-11073-MDSAttributes` shall be used. The value of this DataItem shall be MDS Object 0 attributes (*MDER ASN.1 AttributeList* structure) as defined in IEEE-11073 Part 20601: Application Profile - Optimized Exchange [x] - MDS Attributes. This DataItem shall be returned as a Base64 encoded string.

For MDS Objects 1-n, the DataItem *Type* parameter value `IEEE-11073-MetricAttributes` shall be used. The value of this DataItem shall be MDS Object 1-n attributes (*MDER ASN.1 AttributeList* structure) as defined in IEEE-11073 Part 20601: Application Profile - Optimized Exchange [8] - Metric Attribute**.** This DataItem shall be returned as a Base64 encoded string.

For example:

The DataItem named

$0-Attr

provides a Base-64 encoded string containing the MDS Object Attributes (*MDER ASN.1 AttributeList* structure) for the entire Personal Health Device.

The DataItem named

$1-Attr

provides a Base-64 encoded string containing the Medical Class Object Attributes (*MDER ASN.1 AttributeList* structure) for Handle 1 within the Medical Device System.

#### D.3.1.6 DataItem(s) for IEEE-11073 Medical Object Class Sensors Observations

**Type:** MultiInstance

**Description:** IEEE-11073 SensorURN shall support the following DataItem(s) to report The observations for each Medical Object Class Handle within a IEEE-11073 Medical Device System. The DataItem shall be described by a set of three parameter nodes:

*/UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorURNs/#/SensorURN/DataItems/#/*

*Name*

*Type*

*Encoding*

The following table summarizes the settings for the three parameter values listed above which define named DataItem(s) that return observations from IEEE-11073 Medical Objects.

Note: These Sensor parameters do not directly report sensor data, instead they describe named DataItems which are returned by the sensor using either SOAP actions or transport connections to the sensor.

**Table D.5 — DataItem(s) for Medical Object Class Observations**

| Name | Type | Encoding |
|---|---|---|
| *$[MDS-Handle]-Obs* | *mds:MDSObservation* | *Base64* |

Where: *[MDS-Handle]* is substituted with the Handle value of the corresponding Medical Device System Metric Object. The DataItem *Type* parameter value `IEEE-11073-MDSObservation` shall be used. The value of this DataItem shall be MDS Object 1-n attributes (*MDER ASN.1 ObservationScanList* structure) as defined in IEEE-11073 Part 20601: Application Profile - Optimized Exchange [8]. This DataItem shall be returned as a Base64 encoded string.

For example the DataItem named

$1-Obs

provides a Base-64 encoded string containing the MDS Object observation (*MDER ASN.1 ObservationScanList* structure) for Handle 1 of the IEEE-11073 Personal Health Device.

## D.4 IoT Management and Control for IEEE-11073 Persistent Metric Stores (PM-Store)

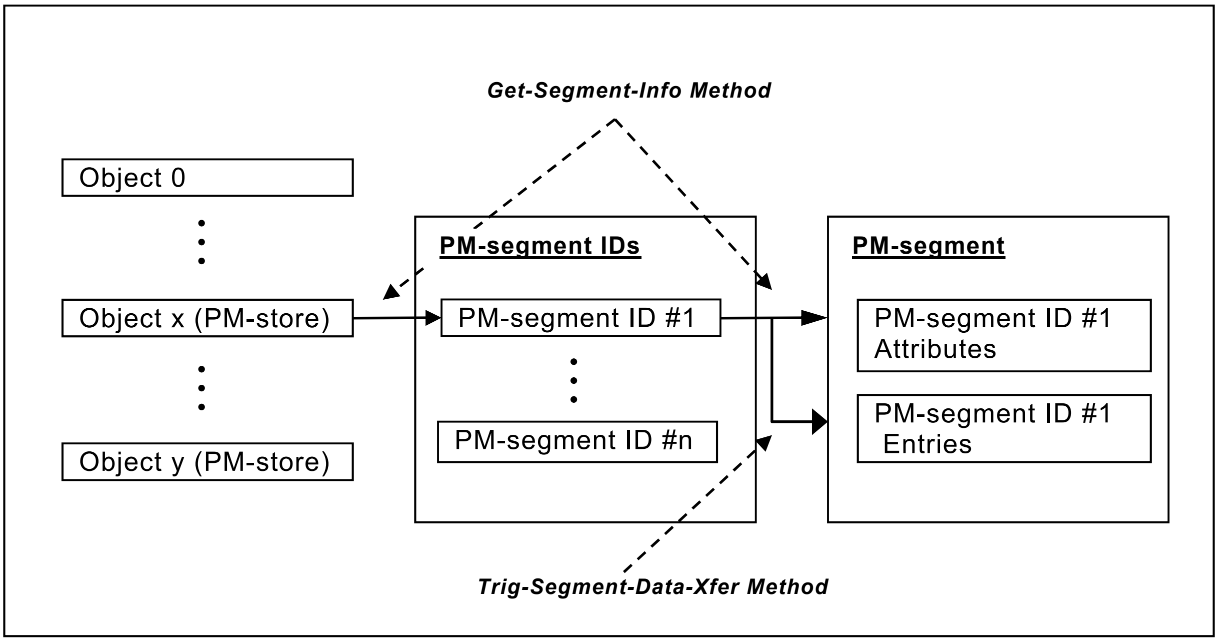

**Figure D.2 — Medical Device System with PM-store Object (informative)**

The UPnP IoT Management and Control device supports IEEE-11073 Persistent Metric Store (PM-store) objects. Each PM-store object supports zero or more PM-segments. Each PM-Segment contains zero of more Entries with each Entry containing an allowed timestamp header and a series of zero of more Elements. Each Element consists of a defined set of observations from one or more Metric Objects within the medical device. PM-segment's attributes define the data format which applies to all Elements within a given PM-segment.

The UPnP IoT Management and Control device allocates a separate *Sensor* node within the UPnP IoT Management and Control DataModel for each PM-store object with a *SensorType* parameter value of "upnp-org:sensors:IEEE-11073_PM-store". This *Sensor* node supports parameters which control operations for the PM-store object such requesting the transfer the current contents of PM-store segments to the UPnP IoT Management and Control device. In addition, each detected PM-segment is allocated a separate *Sensor* node with a *SensorType* parameter value of "upnp-org:sensors:IEEE-11073_PM-segment". Each *Sensor* node corresponding PM-segment(s) supports DataItem(s) to read the attributes of the PM-segment and to identify and read the contents (observations) of each Element in the PM-segment.

The UPnP *IoT Management and Control* device treats each element within a PM-segment as a separate occurrence providing DataItems (*SegEntryIndex* and *SegElementIndex*) which indicate the position of the Element within the PM-segment. When reading the PM-segment sensor all available entries within the PM-segment are sent to the UPnP Client.

### D.4.1 IoT Management and Control Parameters for IEEE-11073 Persistent Metric Stores (PM-Store)

The UPnP IoT Management and Control device represents IEEE-11073 Personal Health Devices as a SensorCollection and an included Sensor. The Sensor shall include the following parameters:

**Table D.6 — IEEE-11073 Persistent Metric Store Parameter Nodes**

| Name | Type | Acc | Req | Reference | EOC | Ver |
|---|---|---|---|---|---|---|

| | | | | | | |
|---|---|---|---|---|---|---|
| UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorsRelated/#/ | MultiInstance | | | | | |
| SensorPath | string | | | D.4.1.2 | | |
| | | | | | | |
| /UPnP/SensorMgt/SensorCollections/#/Sensors/#/ | MultiInstance | | | | | |
| SensorType | string | | | D.4.1.3 | | |
| SensorUpdateRequest | | | | D.4.1.4 | | |
| | | | | | | |

### D.4.1.1 /UPnP/SensorMgt/SensorCollection/#/Sensors/#/SensorRelated/

**Type:** MultiInstance

**Description:** The conditionally required *SensorRelated* MultiInstance node shall be included for IEEE-11073 Medical Device Persistent Metric Store sensors. An instance of this node shall be generated for each PM-segment sensor this PM-store sensor currently supports.

### D.4.1.2 /UPnP/SensorMgt/SensorCollection/#/Sensors/#/SensorRelated/#/SensorPath

**Type:** string

**Description:** The required *SensorPath* parameter shall provide a partial path to a PM-segment sensor supported by this PM-store sensor.

### D.4.1.3 /UPnP/SensorMgt/SensorCollection/#/Sensors/#/SensorType

**Type:** string

**Description:** The following *SensorType* value shall be used for IEEE-11073 Medical Device Persistent Metric Store sensors.

```
PM-store-SensorType ::=  urn:upnp-org:smgt-st:ieee-11073_pm-store
                         :ieee-org:11073_104zzz:<MDS-vendor>:<MDS-model>:telemetry
```

Where: 11073_104zzz indicates the IEEE document number defining the Medical Device Specialization.

### D.4.1.4 /UPnP/SensorMgt/SensorCollection/#/Sensors/#/SensorUpdateRequest

**Type:** boolean

**Description:** The conditionally required *SensorUpdateRequest parameter* shall be implemented for IEEE-11073 PM-store Segment Sensors. When written to (“1”) this parameter shall cause IoT Management and Control to perform the following updates:

- Update the PM-store attributes (as reported by corresponding the *$[PM-store-Handle]-Attr DataItem*).
- Create new PM-Segment Sensor instances for any PM-store segments not already instantiated.

## D.5 IoT Management and Control Parameters for IEEE-11073 Persistent Metric Segments (PM-Segment)

## D.6 IoT Management and Control Parameters for IEEE-11073 Persistent Metric Segments (PM-Segment)

The UPnP IoT Management and Control device represents IEEE-11073 Personal Health Devices as a SensorCollection and an included Sensor. The Sensor shall include the following parameters:

**Table D.7 — IEEE-11073 Persistent Metric Segment Parameter Nodes**

| Name | Type | Acc | Req | Reference | EOC | Ver |
|---|---|---|---|---|---|---|
| UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorsRelated/#/ | MultiInstance | | | | | |
| SensorPath | string | | | D.6.1.3 | | |
| | | | | | | |
| /UPnP/SensorMgt/SensorCollections/#/Sensors/#/ | MultiInstance | | | | | |
| SensorType | string | | | D.6.1.1 | | |
| SensorUpdateRequest | | | | | | |
| | | | | | | |
| /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorURNs/# | MultiInstance | | | | | |
| SensorURN | string | | | D.6.1.5 | | |
| | | | | | | |
| /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorURNs/#/DataItems/# | MultiInstance | | | | | |
| Name | string | | | D.6.1.6 | | |
| Type | string | | | D.6.1.6 | | |
| Encoding | string | | | D.6.1.6 | | |
| | | | | | | |

### D.6.1.1 /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorType

**Type:** string

**Description:** The following *SensorType* value shall be used for IEEE-11073 Medical Device Persistent Metric Segment sensors.

```
PM-segment-SensorType ::=   urn:upnp-org:smgt-st:ieee-11073_pm-segment
                            :ieee-org:11073_104zzz
                            :<MDS-vendor>:<MDS-model>:telemetry
```

Where: 11073_104zzz indicates the IEEE document number defining the Medical Device Specialization.

### D.6.1.2 /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorRelated/

**Type:** MultiInstance

**Description:** The conditionally required *SensorRelated* MultiInstance node shall be included for IEEE-11073 PM-segment sensors. An instance of this node shall be generated for the PM-store sensor that manages this PM-segment sensor.

### D.6.1.3 /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorRelated/#/SensorPath

**Type:** string

**Description:** The required *SensorPath* parameter shall provide a partial path to a PM-store sensor which includes this PM-segment.

### D.6.1.4 /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorUpdateRequest

**Type:** boolean

**Description:** The conditionally required *SensorUpdateRequest* *parameter* shall be implemented for IEEE-11073 PM-store segment sensors. When written to “1” shall cause IoT Management and Control service transfer and retain a copy of the PM-store segment associated with this Sensor. When the transfer is complete this parameter shall be set to “0”.

### D.6.1.5 /UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorURNs/#/SensorURN

**Type:** string

**Description:** The *SensorURN* *parameter* for IEEE-11073 PM-segment Sensors shall use the following value:

```
PM-segment-SensorURN ::= urn:upnp-org:smgt-surn:ieee-11073_pm-segment
                         :ieee-org:11073_104zzz:<MDS-vendor>:<MDS-model>:telemetry
```

Where: 11073_104zzz indicates the IEEE document number defining the Medical Device Specialization [9].

### D.6.1.6 DataItems for IEEE-11073 Persistent Metric Segment Objects

**Type:** MultiInstance

**Description:** The IEEE-11073 SensorURN shall support the following *DataItem(*s) as listed in the table below. Each DataItem shall be described by a set of three parameter nodes:

*/UPnP/SensorMgt/SensorCollections/#/Sensors/#/SensorURNs/#/SensorURN/DataItems/#/*

*Name*

*Type*

*Encoding*

The following table lists the settings for the three parameters above to generate the named DataItem described.

Note: These Sensor parameters do not directly report sensor data, instead they describe named DataItem(s) which are returned by the sensor using either SOAP actions or transport connections to the sensor.

**Table D.8 — IEEE-11073 Persistent Metric Segment DataItem(s)**

| Name | Type | Encoding | Description |
|---|---|---|---|
| *ReceiveTimeStamp* | *xsd:dateTime* | *string* | See Table D.3 |
| *ObservationTimeStamp* | *xsd:dateTime* | *string* | See Table D.3 |
| *PersonID* | *Int* | *string* | See Table D.3 |
| | | | |
| *SegInstanceNumber* | *int* | *string* | The required *SegInstanceNumber* DataItem returns the instance number of the PM-Store segment. |
| *SegEntryIndex* | *int* | *string* | The required *SegEntryIndex* DataItem provides a zero-based index corresponding to this entry within the PM-Store segment.. |
| *SegElementIndex* | *int* | *string* | The required *SegElementIndex* DataItem provides a zero-based index corresponding to this element within the PM-Store segment entry. |
| *SegEntryHeader* | *mds:AbsoluteTime*<br>*mds:RelativeTime*<br>*mds:HighResRelativeTime* | *Base64* | The required *SegEntryHeader* DataItem provides the segment entry header contents which prefix the corresponding PM-Store segment entry. If no segment entry header is defined for this PM-Store segment this entry shall return an empty |

| | | | |
|---|---|---|---|
| | | | string. |
| *SegEntryMap* | *mds:PmSegmentEntryMap* | *Base64* | The required *SegEntryMap* DataItem provides the contents of the PMSegmentEntryMap IEEE-11073 data type which describes the PM-Store entry segment header and element(s). |
| *SegEntryElement* | *mds:SegmentElement* | *Base64* | The required *SegEntryElement* DataItem provides the contents of a single PM-Store segment element. The *SegmentEntryIndex* and *SegmentElementIndex* DataItem(s) shall reflect to indices of this element within the PM-store segment. |

# Annex E Common DataItem Names
# (Normative)

## E.1 Introduction

This annex contains a list of predefined DataItem names. The DataItem naming convention is specified in (see UPnP IoT Management and Control Overview [10], subclause 4.3, "DataItem Semantics" for additional information).

The described DataItems are specified in such way that they will have a default unit. The used unit can be made explicit by using the units of measurement xml section in the DataItem description (see UPnP IoT Management and Control Architecture Overview [10], subclause 4.5 for additional information).

DataModels can have the vector indication “_xyz” as suffix. This means that this value is a vector defined for [X,Y,Z]. The contents will then be then formatted as CSV with the values in order of X,Y,Z.

DataModel names can have the set point indication “_sp” or current value indication “_cur” as suffixes. This is done to when a sensor has latency to achieve a set point value. The value suffixed with “_sp” is the set point value and the value suffixed with “_cur” the current value. When the “_sp” and “_cur” value have the same value then the set point is achieved.

Note that both extensions are possible but should have the order of vector indication followed by setpoint or current value indication.

## E.2 Alphabetical list of the DataItem Names

**Table D-9 Alphabetical list of common DataItem names**

| DataItem Name | Section number |
|---|---|
| *Acceleration* | E.3.1 |
| *Angle* | E.3.2 |
| *Area* | E.3.3 |
| *BatteryState* | E.3.37 |
| *Brightness* | E.3.4 |
| *CleanerState* | E.3.38 |
| *Clock* | E.3.46 |
| *CloseToDevice* | E.3.39 |
| *CODetector* | E.3.47 |
| *CO2Detector* | E.3.47 |
| *ColorRGB* | E.3.5 |

| | |
|---|---|
| ControlValve | E.3.6 |
| Count | E.3.7 |
| Current | E.3.8 |
| Detector | E.3.47 |
| Dewpoint | E.3.48 |
| Direction | E.3.9 |
| Doorbell | E.3.49 |
| DryingState | E.3.40 |
| ElectricalResistance | E.3.54 |
| Energy | E.3.51 |
| EnergyOverloadDetector | E.3.47 |
| Flow | E.3.10 |
| FlowDirection | E.3.11 |
| Force | E.3.50 |
| ForceResistance | E.3.55 |
| Frequency | E.3.12 |
| GasDetector | E.3.47 |
| GlassBreakDetector | E.3.47 |
| GPS | E.3.13 |
| HeatDetector | E.3.47 |
| Humidity | E.3.14 |
| LightDetector | E.3.47 |
| Location | E.3.41 |
| Lock | E.3.42 |
| MagnetoMeter | E.3.43 |
| Mass | E.3.15 |

| | |
|---|---|
| MassFlow | E.3.52 |
| Moisture | E.3.16 |
| MotionDetector | E.3.47 |
| OccupancyDetector | E.3.47 |
| Percentage | E.3.17 |
| Position | E.3.18 |
| Power | E.3.19 |
| PowerSwitch | E.3.20 |
| Precipitation | E.3.53 |
| Presence | E.3.21 |
| Pressure | E.3.22 |
| Proximity | E.3.23 |
| Radiation | E.3.56 |
| RainDetector | E.3.47 |
| RelPosition | E.3.24 |
| RelProximity | E.3.25 |
| RelSpeed | E.3.26 |
| Saturation | E.3.27 |
| SmokeDetector | E.3.47 |
| Speed | E.3.28 |
| SpeedDirection | E.3.29 |
| Switch | E.3.30 |
| Temperature | E.3.31 |
| Ventilation | E.3.57 |
| Vibration | E.3.32 |
| Voltage | E.3.33 |

| | |
|---|---|
| Voltage_dc | E.3.34 |
| Volume | E.3.35 |
| WashProgram | E.3.44 |
| WashState | E.3.45 |
| WaterDetector | E.3.47 |
| WRDirection | E.3.36 |

## E.3 DataItem Definitions

### E.3.1 DataItem Name Acceleration

This Annex describes the dataItem model of the Acceleration actuator/sensor.

#### E.3.1.1 DataItem Acceleration definition

The DataItem is defined with the fields defined in **Table 5-10**.

Acceleration_xyz defines the acceleration with a specific direction.

**Table 5-10 — DataItem Acceleration definition**

| Field | Value | R/A |
|---|---|---|
| Name | Acceleration | R |
| Type | uda:float | R |
| Encoding | ascii | R |
| access | *rw* | R |

[a] For a device this column indicates whether the action shall be implemented or not, where R = required, A = allowed, CR = conditionally required, CA = conditionally allowed, X = Non-standard, add -D when deprecated (e.g., R-D, A-D).

The Acceleration is default expressed in [m/s$^2$].

#### E.3.1.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="Acceleration"
access="rw">
<description>The Acceleration</description>
</DataItemDescription>
```

### E.3.2 DataItem Name Angle

This Annex describes the dataItem model of the Angle actuator/sensor.

### E.3.2.1 DataItem *Angle* definition

The DataItem is defined with the fields defined in **Table 5-11**.

**Table 5-11 — DataItem *Angle* definition**

| Field | Value | R/A |
|---|---|---|
| *Name* | *Angle* | *R* |
| *Type* | uda:float | *R* |
| *Encoding* | *ascii* | *R* |
| *access* | *rw* | *R* |

a For a device this column indicates whether the action shall be implemented or not, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*).

The *Angle* is default expressed in degrees between [0,360].

### E.3.2.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="Angle"
access="rw">
<description>The Angle</description>
</DataItemDescription>
```

### E.3.3 DataItem Name *Area*

This Annex describes the dataItem model of the Area sensor.

### E.3.3.1 DataItem *Area* definition

The sensor is defined with the fields defined in **Table 5-12**.

**Table 5-12 — DataItem *Area* definition**

| Field | Value | R/A |
|---|---|---|
| *Name* | *Area* | *R* |
| *Type* | uda:float | *R* |
| *Encoding* | *ascii* | *R* |
| *access* | *rw* | *R* |

a For a device this column indicates whether the action shall be implemented or not, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*).

The *Area* is default expressed in square meters [m$^2$].

### E.3.3.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
```

```
itemname="Area"
access="rw">
<description>The Area measurement</description>
</DataItemDescription>
```

### E.3.4 DataItem Name Brightness

This Annex describes the dataItem model of the Brightness actuator/sensor.

#### E.3.4.1 DataItem Brightness definition

The sensor is defined with the fields defined in **Table 5-13**.

**Table 5-13 — DataItem Brightness definition**

| Field | Value | R/A |
|---|---|---|
| Name | Brightness | R |
| Type | uda:int32 | R |
| Encoding | ascii | R |
| access | rw | R |
| [a] For a device this column indicates whether the action shall be implemented or not, where R = required, A = allowed, CR = conditionally required, CA = conditionally allowed, X = Non-standard, add -D when deprecated (e.g., R-D, A-D). | | |

The Brightness is used for specifying the brightness of a light/display. This model specifies the intensity on a scale of 0 to 100, with 0 (zero) indicating the minimum intensity.

#### E.3.4.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="Brightness"
access="rw">
<description>The Brightness of a light</description>
</DataItemDescription>
```

### E.3.5 DataItem Name ColorRGB

This Annex describes the dataItem model of the ColorRGB actuator/sensor.

#### E.3.5.1 DataItem ColorRGB definition

The sensor is defined with the fields defined in **Table 5-14**.

**Table 5-14 — DataItem ColorRGB definition**

| Field | Value | R/A |
|---|---|---|
| Name | ColorRGB | R |
| Type | uda:int32 | R |
| Encoding | ascii | R |
| access | rw | R |
| [a] For a device this column indicates whether the action shall be implemented or not, where R = required, A = allowed, CR = conditionally required, CA = conditionally allowed, X = Non-standard, add -D when deprecated (e.g., R-D, A-D). | | |

The RGBColor model is used for specifying colors. This model specifies the intensity of red, green, and blue on a scale of 0 to 255, with 0 (zero) indicating the minimum intensity. The settings of the three colors are converted to a single integer value by using this formula:
RGB value= Red + (Green*256) + (Blue*256*256)

#### E.3.5.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="ColorRGB"
access="rw">
<description>RGB value</description>
</DataItemDescription>
```

### E.3.6 DataItem Name ControlValve

This Annex describes the dataItem model of the ControlValve actuator/sensor.

#### E.3.6.1 DataItem ControlValve definition

The sensor is defined with the fields defined in **Table 5-15**.

**Table 5-15 — DataItem ControlValve definition**

| Field | Value | R/A |
|---|---|---|
| Name | ControlValve | R |
| Type | uda:string | R |
| Encoding | ascii | R |
| access | rw | R |

a For a device this column indicates whether the action shall be implemented or not, where R = required, A = allowed, CR = conditionally required, CA = conditionally allowed, X = Non-standard, add -D when deprecated (e.g., R-D, A-D).

The ControlValve can be expressed as: “open”, “close” and “auto”.

#### E.3.6.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="ControlValve"
access="rw">
<description>The controlvalve value</description>
</DataItemDescription>
```

### E.3.7 DataItem Name Count

This Annex describes the dataItem model of an Count actuator/sensor.

This is an abstracted value that can be used count or set occurrences.

#### E.3.7.1 DataItem Count definition

The sensor is defined with the fields defined in **Table 5-16**.

**Table 5-16 — DataItem *Count* definition**

| Field | Value | R/A |
|---|---|---|
| *Name* | *Count* | *R* |
| *Type* | uda:int32 | *R* |
| *Encoding* | *ascii* | *R* |
| *access* | *rw* | *R* |
| [a] For a device this column indicates whether the action shall be implemented or not, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*). | | |

### E.3.7.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="Count"
access="rw">
<description>counted value</description>
</DataItemDescription>
```

## E.3.8 DataItem Name *Current*

This Annex describes the dataItem model of an Current sensor.

### E.3.8.1 DataItem *Current* definition

The sensor is defined with the fields defined in **Table 5-17**.

**Table 5-17 — DataItem *Current* definition**

| Field | Value | R/A |
|---|---|---|
| *Name* | *Current* | *R* |
| *Type* | uda:float | *R* |
| *Encoding* | *ascii* | *R* |
| *access* | *ro* | *R* |
| [a] For a device this column indicates whether the action shall be implemented or not, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*). | | |

The *Current* is default expressed in Ampere [A].

### E.3.8.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="Current"
access="ro">
<description>Current value</description>
</DataItemDescription>
```

### E.3.9 DataItem Name *Direction*

This Annex describes the dataItem model of an Direction actuator/sensor.

#### E.3.9.1 DataItem *Direction* definition

The sensor is defined with the fields defined in **Table 5-18**.

**Table 5-18 — DataItem *Direction* definition**

| Field | Value | R/A |
|---|---|---|
| Name | Direction | R |
| Type | uda:string | R |
| Encoding | ascii | R |
| access | rw | R |
| a For a device this column indicates whether the action shall be implemented or not, where R = required, A = allowed, CR = conditionally required, CA = conditionally allowed, X = Non-standard, add -D when deprecated (e.g., R-D, A-D). | | |

The *Direction* is expressed with values: “*up*”, ”*down*”, ”*left*”, “*right*”, “*forward*”, “*reverse*” and the values specified for *WRDirection* (see section E.3.36).

#### E.3.9.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="Direction"
access="rw">
<description>Direction measurement</description>
</DataItemDescription>
```

### E.3.10 DataItem Name *Flow*

This Annex describes the dataItem model of an Flow actuator/sensor.

#### E.3.10.1 DataItem *Flow* definition

The sensor is defined with the fields defined in **Table 5-19**.

**Table 5-19 — DataItem *Flow* definition**

| Field | Value | R/A |
|---|---|---|
| Name | Flow | R |
| Type | uda:float | R |
| Encoding | ascii | R |
| access | rw | R |
| a For a device this column indicates whether the action shall be implemented or not, where R = required, A = allowed, CR = conditionally required, CA = conditionally allowed, X = Non-standard, add -D when deprecated (e.g., R-D, A-D). | | |

The *Flow* is default expressed in [$m^3$/s].

#### E.3.10.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
```

xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="*Flow*"
access="rw">
<description>Flow sensor</description>
</DataItemDescription>

### E.3.11 DataItem Name *FlowDirection*

This Annex describes the dataItem model of an FlowDirection actuator/sensor.

#### E.3.11.1 DataItem *FlowDirection* definition

The sensor is defined with the fields defined in **Table 5-20**.

**Table 5-20 — DataItem *FlowDirection* definition**

| Field | Value | R/A |
|---|---|---|
| *Name* | *FlowDirection* | *R* |
| *Type* | uda:string | *R* |
| *Encoding* | *ascii* | *R* |
| *access* | rw | *R* |

[a] For a device this column indicates whether the action shall be implemented or not, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *_X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*).

The *FlowDirection* is expressed as: “*in*” and “*out*”.

#### E.3.11.2 DataItem description xml

<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="*FlowDirection*"
access="rw">
<description>FlowDirection sensor</description>
</DataItemDescription>

### E.3.12 DataItem Name *Frequency*

This Annex describes the dataItem model of an (voltage) Frequency sensor/actuator.

#### E.3.12.1 DataItem *Frequency* definition

The sensor is defined with the fields defined in **Table 5-21**.

**Table 5-21 — DataItem *Frequency* definition**

| Field | Value | R/A |
|---|---|---|
| *Name* | *Frequency* | *R* |
| *Type* | uda:int32 | *R* |
| *Encoding* | *ascii* | *R* |
| *access* | ro | *R* |

| Field | Value | R/A |
|---|---|---|

[a] For a device this column indicates whether the action shall be implemented or not, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*).

The *Frequency* is default expressed in [Hz].

#### E.3.12.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="Frequency"
access="ro">
<description>Frequency</description>
</DataItemDescription>
```

### E.3.13 DataItem Name *GPS*

This Annex describes the dataItem model of an GPS sensor.

#### E.3.13.1 DataItem *GPS* definition

The sensor is defined with the fields defined in **Table 5-22**.

**Table 5-22 — DataItem *GPS* definition**

| Field | Value | R/A |
|---|---|---|
| *Name* | *GPS* | *R* |
| *Type* | uda:string | *R* |
| *Encoding* | *ascii* | *R* |
| *access* | *ro* | *R* |

[a] For a device this column indicates whether the action shall be implemented or not, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*).

The *GPS* coordinate is defined in unit coord-gps or as unit coord-dms section 4.5.6 of [10].

#### E.3.13.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="GPS"
access="ro">
<description>GPS position</description>
</DataItemDescription>
```

### E.3.14 DataItem Name *Humidity*

This Annex describes the dataItem model of an Humidity sensor.

#### E.3.14.1 DataItem *Humidity* definition

The sensor is defined with the fields defined in **Table 5-23**.

**Table 5-23 — DataItem *Humidity* definition**

| Field | Value | R/A |
|---|---|---|
| *Name* | *Humidity* | *R* |
| *Type* | uda:float | *R* |
| *Encoding* | *ascii* | *R* |
| *access* | ro | *R* |

[a] For a device this column indicates whether the action shall be implemented or not, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*).

The *Humidity* is expressed in percentage of water in the atmosphere.

### E.3.14.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="Humidity"
access="ro">
<description>Humidity</description>
</DataItemDescription>
```

## E.3.15 DataItem Name *Mass*

This Annex describes the dataItem model of an Mass sensor.

### E.3.15.1 DataItem *Mass* definition

The sensor is defined with the fields defined in **Table 5-24**.

**Table 5-24 — DataItem *Mass* definition**

| Field | Value | R/A |
|---|---|---|
| *Name* | *Mass* | *R* |
| *Type* | uda:float | *R* |
| *Encoding* | *ascii* | *R* |
| *access* | ro | *R* |

[a] For a device this column indicates whether the action shall be implemented or not, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*).

The *Mass* is default expressed in [kg].

### E.3.15.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="Mass"
access="ro">
<description>Mass</description>
</DataItemDescription>
```

### E.3.16 DataItem Name Moisture

This Annex describes the dataItem model of an Moisture sensor.

#### E.3.16.1 DataItem Moisture definition

The sensor is defined with the fields defined in **Table 5-25**.

**Table 5-25 — DataItem Moisture definition**

| Field | Value | R/A |
|---|---|---|
| Name | Moisture | R |
| Type | uda:float | R |
| Encoding | ascii | R |
| access | ro | R |

[a] For a device this column indicates whether the action shall be implemented or not, where R = required, A = allowed, CR = conditionally required, CA = conditionally allowed, X = Non-standard, add -D when deprecated (e.g., R-D, A-D).

The Moisture is expressed in percentage of water in the measured substance.

#### E.3.16.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="Moisture"
access="ro">
<description>Moisture</description>
</DataItemDescription>
```

### E.3.17 DataItem Name Percentage

This Annex describes the dataItem model of an Percentage actuator/sensor.

#### E.3.17.1 DataItem Percentage definition

The sensor is defined with the fields defined in **Table 5-26**.

**Table 5-26 — DataItem Percentage definition**

| Field | Value | R/A |
|---|---|---|
| Name | Percentage | R |
| Type | uda:int16 | R |
| Encoding | ascii | R |
| access | rw | R |

[a] For a device this column indicates whether the action shall be implemented or not, where R = required, A = allowed, CR = conditionally required, CA = conditionally allowed, X = Non-standard, add -D when deprecated (e.g., R-D, A-D).

The Percentage is used for specifying an value between [0,100] where 0 is the minimum value and 100 is the maximum value.

#### E.3.17.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
```

```
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="Percentage"
access="rw">
<description>Percentage</description>
</DataItemDescription>
```

### E.3.18 DataItem Name *Position*

This Annex describes the dataItem model of an position actuator/sensor.

The generic DataItem Name is defined as *Position*.

The current position is defined as *Position_cur*.

The set point (wanted) is defined as *Position_sp*.

#### E.3.18.1 DataItem *Position* definition

The sensor is defined with the fields defined in **Table 5-27**.

**Table 5-27 — DataItem *Position* definition**

| Field | Value | R/A |
|---|---|---|
| *Name* | *Position* | *R* |
| *Type* | uda:float | *R* |
| *Encoding* | *ascii* | *R* |
| *access* | *rw* | *R* |
| a For a device this column indicates whether the action shall be implemented or not, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*). | | |

The *Position* is default expressed in meters [m].

#### E.3.18.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="Position"
access="rw">
<description>The absolute position</description>
</DataItemDescription>
```

### E.3.19 DataItem Name *Power*

This Annex describes the dataItem model of the Power actuator/sensor.

#### E.3.19.1 DataItem *Power* definition

The sensor is defined with the fields defined in **Table 5-28**.

**Table 5-28 — DataItem *Power* definition**

| Field | Value | R/A |
|---|---|---|

| Field | Value | R/A |
|---|---|---|
| *Name* | *Power* | *R* |
| *Type* | uda:float | *R* |
| *Encoding* | *ascii* | *R* |
| *access* | *rw* | *R* |

[a] For a device this column indicates whether the action shall be implemented or not, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*).

The *Power* is default expressed in watts [W].

### E.3.19.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="Power"
access="rw">
<description>The power</description>
</DataItemDescription>
```

## E.3.20 DataItem Name *PowerSwitch*

This Annex describes the dataItem model of an PowerSwitch actuator/sensor.

### E.3.20.1 DataItem *PowerSwitch* definition

The sensor is defined with the fields defined in **Table 5-29**.

**Table 5-29 — DataItem *PowerSwitch* definition**

| Field | Value | R/A |
|---|---|---|
| *Name* | *PowerSwitch* | *R* |
| *Type* | uda:string | *R* |
| *Encoding* | *ascii* | *R* |
| *access* | *rw* | *R* |

[a] For a device this column indicates whether the action shall be implemented or not, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*).

The *PowerSwitch* is expressed as; “*on*”, “*off*” or “*sleep*”.

### E.3.20.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="PowerSwitch"
access="rw">
<description>The power switch</description>
</DataItemDescription>
```

### E.3.21 DataItem Name *Presence*

This Annex describes the dataItem model of an Presence actuator/sensor.

#### E.3.21.1 DataItem *Presence* definition

The sensor is defined with the fields defined in **Table 5-30**.

**Table 5-30 — DataItem *Presence* definition**

| Field | Value | R/A |
|---|---|---|
| *Name* | *Presence* | *R* |
| *Type* | uda:string | *R* |
| *Encoding* | *ascii* | *R* |
| *access* | *rw* | *R* |

a For a device this column indicates whether the action shall be implemented or not, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*).

The *Presence* is used for specifying presence, and is expressed as “*present*” and “*notpresent*”.

Note that the value “*present”* can also be interpreted as available and the value “*notpresent”* as unavailable.

#### E.3.21.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="Presence"
access="rw">
<description>The presence indication</description>
</DataItemDescription>
```

### E.3.22 DataItem Name *Pressure*

This Annex describes the dataItem model of the Presure actuator/sensor.

The generic DataItem name is defined as *Pressure*.

The current pressure is defined as *Pressure_cur*.

The set point (wanted) presure is defined as *Pressure_sp*.

#### E.3.22.1 DataItem *Pressure* definition

The sensor is defined with the fields defined in **Table 5-31**.

**Table 5-31 — DataItem *Pressure* definition**

| Field | Value | R/A |
|---|---|---|
| *Name* | *Pressure* | *R* |
| *Type* | uda:int32 | *R* |
| *Encoding* | *ascii* | *R* |
| *access* | *rw* | *R* |

| Field | Value | R/A |
|---|---|---|

[a] For a device this column indicates whether the action shall be implemented or not, where R = required, A = allowed, CR = conditionally required, CA = conditionally allowed, X = Non-standard, add -D when deprecated (e.g., R-D, A-D).

The Pressure is default expressed in pascal [Pa].

### E.3.22.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="Pressure"
access="rw">
<description>The pressure of a system</description>
</DataItemDescription>
```

## E.3.23 DataItem Name *Proximity*

This Annex describes the dataItem model of an Proximity actuator/sensor.

### E.3.23.1 DataItem *Proximity* definition

The sensor is defined with the fields defined in**Table 5-32**.

**Table 5-32 — DataItem *Proximity* definition**

| Field | Value | R/A |
|---|---|---|
| Name | Proximity | R |
| Type | uda:float | R |
| Encoding | ascii | R |
| access | rw | R |

[a] For a device this column indicates whether the action shall be implemented or not, where R = required, A = allowed, CR = conditionally required, CA = conditionally allowed, X = Non-standard, add -D when deprecated (e.g., R-D, A-D).

The Proximity is default expressed in meters [m].

### E.3.23.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="Proximity"
access="rw">
<description>Distance measurement</description>
</DataItemDescription>
```

## E.3.24 DataItem Name *RelPosition*

This Annex describes the dataItem model of an relative position actuator/sensor.

The generic DataItem Name is defined as RelPosition.

The current position is defined as RelPosition_cur.

The set point (wanted) position is defined as RelPosition_sp.

#### E.3.24.1 DataItem RelPosition definition

The sensor is defined with the fields defined in **Table 5-33**.

**Table 5-33 — DataItem RelPosition definition**

| Field | Value | R/A |
|---|---|---|
| Name | RelPosition | R |
| Type | uda:float | R |
| Encoding | ascii | R |
| access | rw | R |

[a] For a device this column indicates whether the action shall be implemented or not, where R = required, A = allowed, CR = conditionally required, CA = conditionally allowed, X = Non-standard, add -D when deprecated (e.g., R-D, A-D).

The RelPosition is relative and is expressed in a value between [0,100] where 0 means closed and 100 means fully open.

#### E.3.24.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="RelPosition"
access="rw">
<description>The relative position</description>
</DataItemDescription>
```

### E.3.25 DataItem Name RelProximity

This Annex describes the dataItem model of an RelProximity actuator/sensor.

#### E.3.25.1 DataItem RelProximity definition

The sensor is defined with the fields defined in **Table 5-34**.

**Table 5-34 — DataItem RelProximity definition**

| Field | Value | R/A |
|---|---|---|
| Name | RelProximity | R |
| Type | uda:float | R |
| Encoding | ascii | R |
| access | rw | R |

[a] For a device this column indicates whether the action shall be implemented or not, where R = required, A = allowed, CR = conditionally required, CA = conditionally allowed, X = Non-standard, add -D when deprecated (e.g., R-D, A-D).

The RelProximity (distance) is relative and is expressed in a value between [0,100] where 0 means no distance, and 100 means max distance.

#### E.3.25.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
```

```
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="RelProximity"
access="rw">
<description>The Relative Distance</description>
</DataItemDescription>
```

### E.3.26 DataItem Name *RelSpeed*

This Annex describes the dataItem model of an relative speed actuator/sensor.

The generic DataItem name is defined as*RelSpeed*.

The current speed is defined as *RelSpeed_cur*.

The set point (wanted) speed is defined as *RelSpeed_sp*.

#### E.3.26.1 DataItem *RelSpeed* definition

The sensor is defined with the fields defined in **Table 5-35**.

**Table 5-35 — DataItem *RelSpeed* definition**

| Field | Value | R/A |
|---|---|---|
| *Name* | *RelSpeed* | *R* |
| *Type* | uda:float | *R* |
| *Encoding* | *ascii* | *R* |
| *access* | *rw* | *R* |

[a] For a device this column indicates whether the action shall be implemented or not, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*).

The *RelSpeed* is relative and is expressed in a value between [0,100] where 0 means stationary and 100 means maximum speed.

#### E.3.26.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="RelSpeed"
access="rw">
<description>The relative speed</description>
</DataItemDescription>
```

### E.3.27 DataItem Name *Saturation*

This Annex describes the dataItem model of the Saturation actuator/sensor.

#### E.3.27.1 DataItem *Saturation* definition

The sensor is defined with the fields defined in **Table 5-36**.

**Table 5-36 — DataItem *Saturation* definition**

| Field | Value | R/A |
|---|---|---|
| *Name* | *Saturation* | *R* |
| *Type* | uda:int32 | *R* |
| *Encoding* | *ascii* | *R* |
| *access* | *rw* | *R* |
| [a] For a device this column indicates whether the action shall be implemented or not, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*). | | |

The *Saturation* defines a range from gray (0%) to pure color (100%) at constant lightness level. A pure color is fully saturated.

#### E.3.27.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="Saturation"
access="rw">
<description>The color saturation value</description>
</DataItemDescription>
```

### E.3.28 DataItem Name *Speed*

This Annex describes the dataItem model of the Speed actuator/sensor.

#### E.3.28.1 DataItem *Speed* definition

The sensor is defined with the fields defined in **Table 5-37**.

**Table 5-37 — DataItem *Speed* definition**

| Field | Value | R/A |
|---|---|---|
| *Name* | *Speed* | *R* |
| *Type* | uda:int32 | *R* |
| *Encoding* | *ascii* | *R* |
| *Access* | *rw* | *R* |
| [a] For a device this column indicates whether the action shall be implemented or not, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*). | | |

The *Speed* is default expressed in meters per second [m/s].

#### E.3.28.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="Speed"
access="rw">
<description>The speed value</description>
</DataItemDescription>
```

### E.3.29 DataItem Name *SpeedDirection*

This Annex describes the dataItem model of the speed direction actuator/sensor.

The Generic DataItem name is defined as *SpeedDirection*.

The current speed direction is defined as *SpeedDirection_cur*.

The set point (wanted) speed direction is defined as *SpeedDirection_sp*.

#### E.3.29.1 DataItem *SpeedDirection* definition

The sensor is defined with the fields defined in **Table 5-38**.

**Table 5-38 — DataItem *SpeedDirection* definition**

| Field | Value | R/A |
|---|---|---|
| *Name* | *SpeedDirection* | *R* |
| *Type* | uda:string | *R* |
| *Encoding* | *ascii* | *R* |
| *Access* | *rw* | *R* |

[a] For a device this column indicates whether the action shall be implemented or not, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*).

The *SpeedDirection* (for revolving speeds) is defined as ”*left*” and “*right*”.

#### E.3.29.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="SpeedDirection"
access="rw">
<description>The Direction of the speed</description>
</DataItemDescription>
```

### E.3.30 DataItem Name *Switch*

This Annex describes the dataItem model of the *Switch* sensor.

#### E.3.30.1 DataItem *Switch* definition

The sensor is defined with the fields defined in **Table 5-39**.

**Table 5-39 — DataItem *Switch* definition**

| Field | Value | R/A |
|---|---|---|
| *Name* | *Switch* | *R* |
| *Type* | uda:boolean | *R* |
| *Encoding* | *ascii* | *R* |
| *access* | *rw* | *R* |

[a] For a device this column indicates whether the action shall be implemented or not, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*).

The Swich value "1" means that the device is active, value "0" means that the device is inactive.

### E.3.30.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="Switch"
access="rw">
<description>Generic on/off switch</description>
</DataItemDescription>
```

## E.3.31 DataItem Name Temperature

This Annex describes the dataItem model of the Temperature sensor.

The Temperature sensor definition can occur as Temperature, current Temperature and set point Temperatures

The generic DataItem name is defined as Temperature.

The current temperature is defined as Temperature_cur.

The set point (wanted) temperature is defined as Temperature_sp.

### E.3.31.1 DataItem Temperature definition

The sensor is defined with the fields defined in **Table 5-40**.

**Table 5-40 — DataItem Temperature definition**

| Field | Value | R/A |
|---|---|---|
| Name | Temperature | R |
| Type | uda:int32 | R |
| Encoding | ascii | R |
| access | rw | R |

[a] For a device this column indicates whether the action shall be implemented or not, where R = required, A = allowed, CR = conditionally required, CA = conditionally allowed, X = Non-standard, add -D when deprecated (e.g., R-D, A-D).

The Temperature is default expressed in Celcius [C].

### E.3.31.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="Temperature"
access="rw">
<description>The generic, current or set point temperature</description>
</DataItemDescription>
```

### E.3.32 DataItem Name Vibration

This Annex describes the dataItem model of the vibration actuator/sensor.

#### E.3.32.1 DataItem Vibration definition

The sensor is defined with the fields defined in **Table 5-41**.

**Table 5-41 — DataItem Vibration definition**

| Field | Value | R/A |
|---|---|---|
| Name | Vibration | R |
| Type | uda:float | R |
| Encoding | ascii | R |
| Access | rw | R |

[a] For a device this column indicates whether the action shall be implemented or not, where R = required, A = allowed, CR = conditionally required, CA = conditionally allowed, X = Non-standard, add -D when deprecated (e.g., R-D, A-D).

The Vibration is default expressed in [Hz].

#### E.3.32.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="Vibration"
access="rw">
<description>The vibration of a system</description>
</DataItemDescription>
```

### E.3.33 DataItem Name Voltage

This Annex describes the dataItem model of the alternating voltage actuator/sensor.

The generic DataItem name is defined as Voltage.

The voltage of the different phases are defined as Voltage_phX, where X is defined as 1,2 and 3.

#### E.3.33.1 DataItem Voltage definition

The sensor is defined with the fields defined in **Table 5-42**.

**Table 5-42 — DataItem Voltage definition**

| Field | Value | R/A |
|---|---|---|
| Name | Voltage | R |
| Type | uda:float | R |
| Encoding | ascii | R |
| Access | rw | R |

[a] For a device this column indicates whether the action shall be implemented or not, where R = required, A = allowed, CR = conditionally required, CA = conditionally allowed, X = Non-standard, add -D when deprecated (e.g., R-D, A-D).

The Voltage is default expressed in Volts [v].

### E.3.33.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="Voltage"
access="rw">
<description>The voltage of a system</description>
</DataItemDescription>
```

## E.3.34 DataItem Name Voltage_dc

This Annex describes the dataItem model of the direct current voltage actuator/sensor.

### E.3.34.1 DataItem Voltage_dc definition

The sensor is defined with the fields defined in **Table 5-43**.

**Table 5-43 — DataItem Voltage_dc definition**

| Field | Value | R/A |
|---|---|---|
| Name | Voltage_dc | R |
| Type | uda:float | R |
| Encoding | ascii | R |
| access | rw | R |

[a] For a device this column indicates whether the action shall be implemented or not, where R = required, A = allowed, CR = conditionally required, CA = conditionally allowed, X = Non-standard, add -D when deprecated (e.g., R-D, A-D).

The Voltage_dc is default expressed in Volts [v].

### E.3.34.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="Voltage_dc"
access="rw">
<description>The dc voltage of a system</description>
</DataItemDescription>
```

## E.3.35 DataItem Name Volume

This Annex describes the dataItem model of the volume sensor.

### E.3.35.1 DataItem Volume definition

The sensor is defined with the fields defined in **Table 5-44**.

**Table 5-44 — DataItem Volume definition**

| Field | Value | R/A |
|---|---|---|

| Field | Value | R/A |
|---|---|---|
| *Name* | *Volume* | *R* |
| *Type* | uda:float | *R* |
| *Encoding* | *ascii* | *R* |
| *access* | ro | *R* |

[a] For a device this column indicates whether the action shall be implemented or not, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*).

The *Volume* is default expressed in [m$^3$].

### E.3.35.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="volume"
access="ro">
<description>The volume</description>
</DataItemDescription>
```

## E.3.36 DataItem Name *WRDirection*

This Annex describes the dataItem model of an Windrose Direction sensor.

### E.3.36.1 DataItem *WRDirection* definition

The sensor is defined with the fields defined in **Table 5-45**.

**Table 5-45 — DataItem *WRDirection* definition**

| Field | Value | R/A |
|---|---|---|
| *Name* | *WRDirection* | *R* |
| *Type* | uda:string | *R* |
| *Encoding* | *ascii* | *R* |
| *access* | rw | *R* |

[a] For a device this column indicates whether the action shall be implemented or not, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*).

The *WRDirection* (wind rose direction) is expressed in (starting from North clock wise direction):

“*N*”

“*NNE*”

“*NE*”

“*ENE*”

“*E*”

“*ESE*”

“*SE*”

"SSE"

"S"

"SSW"

"SW"

"WSW"

"W"

"WNW"

"NW"

"NNW"

#### E.3.36.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="WRDirection"
access="rw">
<description>Wind Rose direction</description>

</DataItemDescription>
```

### E.3.37 DataItem Name BatteryState

This Annex describes the dataItem model of the  BatteryState sensor.

#### E.3.37.1 DataItem BatteryState definition

The DataItem is defined with the fields defined in **Table 5-46**.

**Table 5-46 — DataItem BatteryState definition**

| Field | Value | R/A |
|---|---|---|
| Name | BatteryState | R |
| Type | uda:int | R |
| Encoding | ascii | R |
| access | rw | R |
| a For a device this column indicates whether the action shall be implemented or not, where R = required, A = allowed, CR = conditionally required, CA = conditionally allowed, X = Non-standard, add -D when deprecated (e.g., R-D, A-D). | | |

The Batterystate is expressed in percentage, where 0 means empty battery and 100 means full battery.

#### E.3.37.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
```

```
itemname="BatteryState"
access="rw">
<description>The Battery State</description>
</DataItemDescription>
```

### E.3.38 DataItem Name CleanerState

This Annex describes the dataItem model of the Cleaner actuator/sensor.

#### E.3.38.1 DataItem CleanerState definition

The DataItem is defined with the fields defined in **Table 5-47**.

**Table 5-47 — DataItem CleanerState definition**

| Field | Value | R/A |
|---|---|---|
| Name | CleanerState | R |
| Type | uda:string | R |
| Encoding | ascii | R |
| access | rw | R |
| a For a device this column indicates whether the action shall be implemented or not, where R = required, A = allowed, CR = conditionally required, CA = conditionally allowed, X = Non-standard, add -D when deprecated (e.g., R-D, A-D). | | |

The CleanerState is expressed as “Start”, ”Homing”, “Cleaning”, “Charging”, “Completed”.

#### E.3.38.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="CleanerState"
access="rw">
<description>The Cleaner State</description>
</DataItemDescription>
```

### E.3.39 DataItem Name CloseToDevice

This Annex describes the dataItem model of the close to another device location indication.

This value describes if this device is close to another device. This other device might have a known location.

#### E.3.39.1 DataItem CloseToDevice definition

The DataItem is defined with the fields defined in **Table 5-48**.

Table 5-48 — DataItem *CloseToDevice* definition

| Field | Value | R/A |
|---|---|---|
| *Name* | *CloseToDevice* | *R* |
| *Type* | uda:string | *R* |
| *Encoding* | *ascii* | *R* |
| *access* | rw | *R* |
| [a] For a device this column indicates whether the action shall be implemented or not, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*). | | |

The *CloseToDevice* indicates another device id.

#### E.3.39.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="CloseToDevicee"
access="rw">
<description>The ID of another device that is close to this one
</description>
</DataItemDescription>
```

### E.3.40 DataItem Name *DryingState*

This Annex describes the dataItem model of the Lock actuator/sensor.

#### E.3.40.1 DataItem *DryingState* definition

The DataItem is defined with the fields defined in **Table 5-49**.

Table 5-49 — DataItem *DryingState* definition

| Field | Value | R/A |
|---|---|---|
| *Name* | *DryingState* | *R* |
| *Type* | uda:string | *R* |
| *Encoding* | *ascii* | *R* |
| *access* | rw | *R* |
| [a] For a device this column indicates whether the action shall be implemented or not, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*). | | |

The *DryingState* is default expressed as “Idle”, “Drying”, “Completed”.

#### E.3.40.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="DryingState"
access="rw">
```

```
<description>The Drying state</description>
</DataItemDescription>
```

### E.3.41 DataItem Name Location

This Annex describes the dataItem model of the Location indication.

This value describes the location of an device as dataItem. The syntax is the same as described in Annex F.

#### E.3.41.1 DataItem Location definition

The DataItem is defined with the fields defined in **Table 5-50**.

**Table 5-50 — DataItem Location definition**

| Field | Value | R/A |
|---|---|---|
| Name | Location | R |
| Type | uda:string | R |
| Encoding | ascii | R |
| access | rw | R |

[a] For a device this column indicates whether the action shall be implemented or not, where R = required, A = allowed, CR = conditionally required, CA = conditionally allowed, X = Non-standard, add -D when deprecated (e.g., R-D, A-D).

The Location indicates the location of the device.

#### E.3.41.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="Location"
access="rw">
<description>The Location of the device</description>
</DataItemDescription>
```

### E.3.42 DataItem Name Lock

This Annex describes the dataItem model of the Lock actuator/sensor.

#### E.3.42.1 DataItem Lock definition

The DataItem is defined with the fields defined in **Table 5-51**.

The DataItem name can be prefixed with:

- “Door”
- “Window”
- Device types that can be locked, full list is defined in Annex C. Examples:
  - Dishwasher
  - Washing_machine
  - Dryer

- Microwave

Table 5-51 — DataItem Lock definition

| Field | Value | R/A |
|---|---|---|
| Name | Lock | R |
| Type | uda:string | R |
| Encoding | ascii | R |
| access | rw | R |
| a For a device this column indicates whether the action shall be implemented or not, where R = required, A = allowed, CR = conditionally required, CA = conditionally allowed, X = Non-standard, add -D when deprecated (e.g., R-D, A-D). | | |

The Lock is default expressed as “locked”, “unlocked”.

#### E.3.42.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="Lock"
access="rw">
<description>The Lock</description>
</DataItemDescription>
```

### E.3.43 DataItem Name MagnetoMeter

This Annex describes the dataItem model of the Lock actuator/sensor.

#### E.3.43.1 DataItem MagnetoMeter definition

The DataItem is defined with the fields defined in **Table 5-52**.

.

Table 5-52 — DataItem MagnetoMeter definition

| Field | Value | R/A |
|---|---|---|
| Name | MagnetoMeter | R |
| Type | uda:float | R |
| Encoding | ascii | R |
| access | rw | R |
| a For a device this column indicates whether the action shall be implemented or not, where R = required, A = allowed, CR = conditionally required, CA = conditionally allowed, X = Non-standard, add -D when deprecated (e.g., R-D, A-D). | | |

The MagnetoMeter is default expressed in Tesla.

#### E.3.43.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
```

```
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="MagnetoMeter"
access="rw">
<description>The MagnetoMeter</description>
</DataItemDescription>
```

### E.3.44 DataItem Name WashProgram

This Annex describes the dataItem model of the WashProgram actuator/sensor.

#### E.3.44.1 DataItem WashProgram definition

The DataItem is defined with the fields defined in **Table 5-53**.

The WashProgram can be used in a WashingMachine or Dishwasher.

Note that not all states are applicable for a Dishwasher.

**Table 5-53 — DataItem WashProgram definition**

| Field | Value | R/A |
|---|---|---|
| Name | WashProgram | R |
| Type | uda:float | R |
| Encoding | ascii | R |
| access | rw | R |
| a For a device this column indicates whether the action shall be implemented or not, where R = required, A = allowed, CR = conditionally required, CA = conditionally allowed, X = Non-standard, add -D when deprecated (e.g., R-D, A-D). | | |

The WashProgram is expressed as “PreWash”, “FastWash”, “NormalWash”, “IntensiveWash”, “Rinse”, ”Spin”, “Rinse&Spin”, “White”, ”Color”, “Fine”, “Cotton”, “Wool”, “Synthetic” “PumpOut”. The WashProgram definitions can be extended by vendors. The vendor defined states shall start with “X_”.

#### E.3.44.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="WashProgram"
access="rw">
<description>The Wash program</description>
</DataItemDescription>
```

### E.3.45 DataItem Name WashState

This Annex describes the dataItem model of the WashState actuator/sensor.

#### E.3.45.1 DataItem WashState definition

The DataItem is defined with the fields defined in **Table 5-54**.

The WashState can be used in a WashingMachine or Dishwasher.

Note that the “Spinning” state in a Dishwasher can be omitted.

**Table 5-54 — DataItem WashState definition**

| Field | Value | R/A |
|---|---|---|
| Name | WashState | R |
| Type | uda:float | R |
| Encoding | ascii | R |
| access | rw | R |
| a For a device this column indicates whether the action shall be implemented or not, where R = required, A = allowed, CR = conditionally required, CA = conditionally allowed, X = Non-standard, add -D when deprecated (e.g., R-D, A-D). | | |

The WashState is expressed as “Idle”, ”Washing”, “Rinsing”, “Spinning”, “Drying”, “Completed”.

#### E.3.45.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="WashState"
access="rw">
<description>The WashState</description>
</DataItemDescription>
```

### E.3.46 DataItem Name Clock

This Annex describes the dataItem model of a Clock sensor.

#### E.3.46.1 DataItem Clock definition

The DataItem is defined with the fields defined in **Table 5-55**.

**Table 5-55 — DataItem Clock definition**

| Field | Value | R/A |
|---|---|---|
| Name | Clock | R |
| Type | uda:datetime | R |
| Encoding | ascii | R |
| access | rw | R |
| a For a device this column indicates whether the action shall be implemented or not, where R = required, A = allowed, CR = conditionally required, CA = conditionally allowed, X = Non-standard, add -D when deprecated (e.g., R-D, A-D). | | |

The Clock is expressed in uda:daytime.

#### E.3.46.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="Clock"
```

```
access="rw">
<description>The current date and time</description>
</DataItemDescription>
```

### E.3.47 DataItem Name Detector

This Annex describes the dataItem model of the Detector sensor.

Detectors will be prefixed with one of:

- CO (Carbon monoxide)
  - DataItem name: CODetector
- CO2 (Carbon dioxide)
  - DataItem name: CO2Detector
- EnergyOverload (circuit breaker)
  - DataItem name: EnergyOverloadDetector
- Gas (for usage to detect flammable/explosive gasses)
  - DataItem name: GasDetector
- GlassBreak (to detect if an fire alarm is activated)
  - DataItem name: GlassBreakDetector
- Heat (for usage to detect fire)
  - DataItem name: HeatDetector
- Light (to detect the a threshold for an luminance)
  - DataItem name: LightDetector
- Motion (to detect if motion is detected in a room/venue)
  - DataItem name: MotionDetector
- Occupancy (to detect if people are in the room/venue)
  - DataItem name: OccupancyDetector
- Water
  - DataItem name: WaterDetector
- Rain
  - DataItem name: RainDetector
- Smoke
  - DataItem name: SmokeDetector

Note that additional detectors may be added in a later version of the specification.

#### E.3.47.1 DataItem *Detector* definition

The DataItem is defined with the fields defined in **Table 5-56**.

**Table 5-56 — DataItem *Detector* definition**

| Field | Value | R/A |
|---|---|---|
| *Name* | *Detector* | *R* |
| *Type* | *uda:boolean* | *R* |
| *Encoding* | *ascii* | *R* |
| *access* | rw | *R* |
| a For a device this column indicates whether the action shall be implemented or not, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*). | | |

The *Detector* is expressed as boolean where true means detected and false means not detected.

#### E.3.47.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="Detector"
access="rw">
<description>The Detector</description>
</DataItemDescription>
```

### E.3.48 DataItem Name Dewpoint

This Annex describes the dataItem model of the Dewpoint sensor. The dewpoint is the temperature at which the water vapor in a sample of air at constant barometric pressure condenses into liquid water at the same rate at which it evaporates. At temperatures below the dew point, water will leave the air

#### E.3.48.1 DataItem *Dewpoint* definition

The DataItem is defined with the fields defined in **Table 5-57**.

**Table 5-57 — DataItem *Dewpoint* definition**

| Field | Value | R/A |
|---|---|---|
| *Name* | *Dewpoint* | *R* |
| *Type* | *uda:float* | *R* |
| *Encoding* | *ascii* | *R* |
| *access* | rw | *R* |
| a For a device this column indicates whether the action shall be implemented or not, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*). | | |

The *Dewpoint* is expressed in degrees kelvin.

#### E.3.48.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="Dewpoint"
access="rw">
<description>The dewpoint</description>
</DataItemDescription>
```

### E.3.49 DataItem Name Doorbell

This Annex describes the dataItem model of the Doorbell actuator.

#### E.3.49.1 DataItem Doorbell definition

The DataItem is defined with the fields defined in **Table 5-58**.

**Table 5-58 — DataItem Doorbell definition**

| Field | Value | R/A |
|---|---|---|
| Name | Doorbell | R |
| Type | uda:boolean | R |
| Encoding | ascii | R |
| access | rw | R |

[a] For a device this column indicates whether the action shall be implemented or not, where R = required, A = allowed, CR = conditionally required, CA = conditionally allowed, X = Non-standard, add -D when deprecated (e.g., R-D, A-D).

The Doorbell is expressed as boolean where true means producing a sound and false means silence.

#### E.3.49.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="Doorbell"
access="rw">
<description>The Doorbell</description>
</DataItemDescription>
```

### E.3.50 DataItem Name Force

This Annex describes the dataItem model of the Force sensor.

Force_xyz defines the force with a specific direction.

#### E.3.50.1 DataItem Force definition

The DataItem is defined with the fields defined in **Table 5-59**.

Table 5-59 — DataItem *Force* definition

| Field | Value | R/A |
|---|---|---|
| *Name* | *Force* | *R* |
| *Type* | *uda:float* | *R* |
| *Encoding* | *ascii* | *R* |
| *access* | rw | *R* |
| a For a device this column indicates whether the action shall be implemented or not, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*). | | |

The *Force* is expressed in [Nm].

### E.3.50.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="Force"
access="rw">
<description>The Force</description>
</DataItemDescription>
```

## E.3.51 DataItem Name *Energy*

This Annex describes the dataItem model of the energy sensor.

### E.3.51.1 DataItem Energy definition

The DataItem is defined with the fields defined in **Table 5-60**.

Table 5-60 — DataItem *Energy* definition

| Field | Value | R/A |
|---|---|---|
| *Name* | *Energy* | *R* |
| *Type* | *uda:float* | *R* |
| *Encoding* | *ascii* | *R* |
| *access* | rw | *R* |
| a For a device this column indicates whether the action shall be implemented or not, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*). | | |

The *Energy* is expressed in [kwh].

### E.3.51.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="Energy"
access="rw">
```

```
<description>The Energy</description>
</DataItemDescription>
```

### E.3.52 DataItem Name Massflow

This Annex describes the dataItem model of the Massflow sensor.

#### E.3.52.1 DataItem Massflow definition

The DataItem is defined with the fields defined in **Table 5-61**.

**Table 5-61 — DataItem Massflow definition**

| Field | Value | R/A |
|---|---|---|
| Name | Massflow | R |
| Type | uda:float | R |
| Encoding | ascii | R |
| access | rw | R |

[a] For a device this column indicates whether the action shall be implemented or not, where R = required, A = allowed, CR = conditionally required, CA = conditionally allowed, X = Non-standard, add -D when deprecated (e.g., R-D, A-D).

The Massflow is expressed in [kg/s].

#### E.3.52.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="Massflow"
access="rw">
<description>The Massflow</description>
</DataItemDescription>
```

### E.3.53 DataItem Name Precipitation

This Annex describes the dataItem model of the Rain sensor.

#### E.3.53.1 DataItem Precipitation definition

The DataItem is defined with the fields defined in **Table 5-62**.

**Table 5-62 — DataItem Precipitation definition**

| Field | Value | R/A |
|---|---|---|
| Name | Precipitation | R |
| Type | uda:float | R |
| Encoding | ascii | R |
| access | rw | R |

[a] For a device this column indicates whether the action shall be implemented or not, where R = required, A = allowed, CR = conditionally required, CA = conditionally allowed, X = Non-standard, add -D when deprecated (e.g., R-D, A-D).

The *Precipitation* is expressed in [m], note that zero means that is it not precipitation measured. Note that precipitation is expressed in UIs in millimeters or inches (country dependent).

#### E.3.53.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="Precipitation"
access="rw">
<description>The precipitation detector including rate</description>
</DataItemDescription>
```

### E.3.54 DataItem Name *ElectricalResistance*

This Annex describes the dataItem model of the Electrical Resistance sensor.

#### E.3.54.1 DataItem *ElectricalResistance* definition

The DataItem is defined with the fields defined in **Table 5-63**.

**Table 5-63 — DataItem *ElectricalResistance* definition**

| Field | Value | R/A |
|---|---|---|
| *Name* | *ElectricalResistance* | *R* |
| *Type* | *uda:float* | *R* |
| *Encoding* | *ascii* | *R* |
| *access* | rw | *R* |

[a] For a device this column indicates whether the action shall be implemented or not, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*).

The *ElectricalResistance* is expressed in [ohm].

#### E.3.54.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="ElectricalResistance"
access="rw">
<description>The Electrical Resistance</description>
</DataItemDescription>
```

### E.3.55 DataItem Name *ForceResistance*

This Annex describes the dataItem model of the Force Resistance sensor.

#### E.3.55.1 DataItem *ForceResistance* definition

The DataItem is defined with the fields defined in **Table 5-64**.

**Table 5-64 — DataItem ForceResistance definition**

| Field | Value | R/A |
|---|---|---|
| Name | ForceResistance | R |
| Type | uda:float | R |
| Encoding | ascii | R |
| access | rw | R |

[a] For a device this column indicates whether the action shall be implemented or not, where R = required, A = allowed, CR = conditionally required, CA = conditionally allowed, X = Non-standard, add -D when deprecated (e.g., R-D, A-D).

The ForceResistance is expressed in [N].

#### E.3.55.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="ForceResistance"
access="rw">
<description>The Force Resistance</description>
</DataItemDescription>
```

### E.3.56 DataItem Name Radiation

This Annex describes the dataItem model of the Radiation sensor.

Radiation can be prefixed with:

- Alpha (for Alpha radiation)
- Gamma (for Gamma radiation)
- InfraRed (for InfraRed radiation)
- Rontgen (for Rontgen radiation)
- Solar (for solar radiation)
- UltraViolet (for UV radiation)

#### E.3.56.1 DataItem Radiation definition

The DataItem is defined with the fields defined in **Table 5-65**.

Table 5-65 — DataItem *Radiation* definition

| Field | Value | R/A |
|---|---|---|
| *Name* | *Radiation* | *R* |
| *Type* | *uda:float* | *R* |
| *Encoding* | *ascii* | *R* |
| *access* | rw | *R* |

a For a device this column indicates whether the action shall be implemented or not, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*).

The *Radiation* is expressed in [mRad].

#### E.3.56.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="Radiation"
access="rw">
<description>The Radition</description>
</DataItemDescription>
```

### E.3.57 DataItem Name *Ventilation*

This Annex describes the dataItem model of the Ventilation sensor/actuator.

#### E.3.57.1 DataItem *Ventilation* definition

The DataItem is defined with the fields defined in **Table 5-66**.

Table 5-66 — DataItem *Ventilation* definition

| Field | Value | R/A |
|---|---|---|
| *Name* | *Ventilation* | *R* |
| *Type* | *uda:int* | *R* |
| *Encoding* | *ascii* | *R* |
| *access* | rw | *R* |

a For a device this column indicates whether the action shall be implemented or not, where *R* = required, *A* = allowed, *CR* = conditionally required, *CA* = conditionally allowed, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *A-D*).

The *Ventilation* is expressed in percentage where 0% means no ventilation and 100% means maximum ventilation.

#### E.3.57.2 DataItem description xml

```
<DataItemDescription
xmlns="urn:schemas-upnp-org:smgt:sdmdid"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation=".
urn:schemas-upnp-org:smgt:sdmdid.xsd
http://www.upnp.org/schemas/smgt/sdmdid-v1.xsd"
itemname="Ventilation"
```

```
access="rw">
<description>The ventilation</description>
</DataItemDescription>
```

# Annex F Location

This Annex describes the EBNF of the location.

The location can be used to identify where a sensor/actuator is located

Currently there are 2 mechanisms defining the location of a Device; the prefix of the DataItem and the Location as a DataItem.

The mechanism to prefix the data items is described in [10]– section 4.3.2],. The EBNF of the location contains characters to distinguish between free text and the EBNF location. When using the EBNF notation the restricted characters specified in [10]– section 4.3.2] are allowed.

The EBNF location syntax can also be used in a Location DataItem. This DataItem is specified to indicate the location of a Device by means of the syntax of F.1.

## F.1 Location syntax

```
Location          ::=  buildinglocation            |
                       vehicle                     |
                       gpslocation                 |
                       freetext                    |

buildinglocation  ::=  '#' [site_][buildinglist '_'] roomlist
                       [('_')absinteger][('_')relpos]
vehicle           ::=  '?'vehiclelist

gpslocation       ::=  '%dms' dmscoord | '%gps' gpscoord
freetext          ::=  (a-z,A-Z)+
                       (* anyfree text *)

dmscoord          ::=  'lat:'deg','min','sec',lon:'deg','min','sec
gpscoord          ::=  'lat:'float', lon:'float[',alt:'float]
                       (* altitude in meters above sealevel*)
relpos            ::= '{'float','float','float'}'
                       (* relative postion in the room, defined in abstract
                       box of [-1,-1,-1],[1,1,1], [width,length,height] where
                       [0,0,0] is the center of the room, the shortest wall of
                       the room is the width of the room *)

deg               ::=  integer
min               ::=  integer
sec               ::=  integer
float             ::=  [-](1-9)[(0-9)+][.(0-9)+]
integer           ::=  [-](1-9)[(0-9)+]
absinteger        ::=  (1-9)[(0-9)+]


site              ::=  'campus'                    |
                       'city'                      |
                       'technologypark'            |
                       'town'

buildinglist      ::=  'airport'                   |
                       'arena'                     |
                       'bar'                       |
                       'bar'                       |
                       'barn'                      |
                       'brewery'                   |
                       'business'                  |
                       'busstation'                |
                       'cafe'                      |
                       'chalet'                    |
                       'church'                    |
                       'club'                       |
                       'conferencecenter'           |
                       'court'                      |
                       'gasstation'                 |
                       'gym'                        |
                       'factory'                    |
                       'ferryterminal'              |
                       'firestation'                |
                       'home'                       |
                       'holidayhome'                |
                       'hospital'                   |
                       'hotel'                      |
                       'mall'                       |
                       'monastery'                  |
                       'mosque'                     |
                       'motel'                      |
                       'museum'                     |
                       'office'                     |
                       'operahouse'                 |
                       'parkinggarage'              |
                       'policestation'              |
                       'port'                       |
                       'prison'
                       'railwaystation'             |
                       'restaurant'                 |
                       'school'                     |
                       'shop'                       |
                       'stadium'                    |
                       'supermarket'                |
                       'synagogue'                  |
                       'teahouse'                   |
                       'temple'                     |
                       'terminal'                   |
                       'theater'                    |
                       'townhall'                   |
                       'treehouse'                  |
                       'winery'


roomlist          ::=  'attic'                      |
                       'balcony'                    |
                       'ballroom'                   |
                       'bathroom'                   |
                       'bedroom'                    |
                       'border'                     |
                       'boxroom'                    |
                       'cellar'                     |
                       'cloakroom'                  |
                       'conservatory'               |
                       'corridor'                   |
                       'deck'                       |
                       'den'                        |
                       'diningroom'                 |
                       'drawingroom'                |
                       'driveway'                   |
                       'dungeon'                    |
                       'ensuite'                    |
                       'entrance'                   |
                       'familyroom'                 |
                       'garage'                     |
                       'garden'                     |
                       'guestroom'                  |
                       'hall'                       |
                       'kitchen'                    |
                       'larder'                     |
                       'lawn'                       |
                       'library'                    |
                       'livingroom'                 |
                        'lounge'                     |
                        'mancave'                    |
                        'masterbedroom'              |
                        'musicroom'                  |
                        'office'                     |
                        'pantry'                     |
                        'parkinglot'                 |
                        'parlour'                    |
                        'patio'                      |
                        'receiptionroom'             |
                        'roof'                       |
                        'roofterrace'                |
                        'sauna'                      |
                        'shed'                       |
                        'sittingroom'                |
                        'snug'                       |
                        'spa'                        |
                        'studio'                     |
                        'suite'                      |
                        'swimmingpool'               |
                        'toilet'                     |
                        'utilityroom'                |
                        'ward'                       |
                        'vegitableplot'              |
                        'terrace'                    |
                        'utilityroom'                |


vehiclelist ::=         'airplane'                   |
                        'bicycle'                    |
                        'boat'                       |
                        'bus'                        |
                        'car'                        |
                        'cruiseship'                 |
                        'ferry'                      |
                        'hoverboard'                 |
                        'motorcycle'                 |
                        'motorhome'                  |
                        'ship'                       |
                        'spaceship'                  |
                        'train'                      |
                        'yacht'
```

## F.2 Example of standard names

#bathroom

#bedroom_2

#livingroom_{0,0.5,1}

%gpslat:-1.3535400 ,lon:50.9684900

%dmslat:37,15 ,47,lon:115,47,34

**ICS 35.200**

Price based on 105 pages