



BSI Standards Publication

Information technology — UPnP Device Architecture

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

National foreword

This British Standard is the UK implementation of ISO/IEC 29341-30-11:2017.

The UK participation in its preparation was entrusted to Technical Committee ICT/-/1, Information systems co-ordination.

A list of organizations represented on this committee can be obtained on request to its secretary.

This publication does not purport to include all the necessary provisions of a contract. Users are responsible for its correct application.

© The British Standards Institution 2017
Published by BSI Standards Limited 2017

ISBN 978 0 580 90904 7

ICS 35.200

Compliance with a British Standard cannot confer immunity from legal obligations.

This British Standard was published under the authority of the Standards Policy and Strategy Committee on 31 July 2017.

Amendments/corrigenda issued since publication

Date	Text affected
------	---------------

**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)

AVA Documents
Document Sharing Hub
© ISO/IEC 2017



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2017, Published in Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
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

1	Scope.....	viii
2	Normative References	1
3	Terms, Definitions and Abbreviations	2
4	Notations and conventions	2
4.1	Notation	2
4.2	Data Types	3
4.3	Vendor-defined Extensions	3
4.3.1	Extended Backus-Naur Form	3
5	IoT Management and Control Configuration Management Service Profile	4
5.1	Service Type	4
5.2	IoT Management and Control Device Architecture	4
5.2.1	Sensor Description	4
5.2.2	Sensor Data Interface	4
5.2.3	Sensor Data Records	4
5.2.4	SensorURNs	4
5.2.5	Sensor Required DataItems	4
5.2.6	Sensor Normative Type Identifiers	5
5.2.7	Event Model	6
5.3	State Variables	7
5.4	Actions	8
5.4.1	Introduction	8
5.4.2	<u>GetSupportedDataModels()</u>	9
5.4.3	<u>GetSelectedValues()</u>	10
5.4.4	<u>SetValues()</u>	10
5.4.5	<u>CreateInstance()</u>	10
5.4.6	<u>DeleteInstance()</u>	10
5.4.7	<u>SetAttributes()</u>	10
	Annex A IoT Management and Control General Data Model (normative)	11
	Annex B Required IoT Management and Control DataItem(s) (normative)	38
	Annex C Common Device Identifiers (normative)	39
	Annex D IEEE-11073 Personal Health Devices	49
	Figure A.1 — Script status state machine	26
	Figure A.2 — Sample Collection	31
	Figure A.3 — Sample of a GroupSet	34
	Figure A.4 — Sample of receiving the Brightness setting	35
	Figure D.1 — Blood Pressure Monitor – Medical Device System (Informative)	49
	Figure D.2 — Medical Device System with PM-store Object (informative)	57
	Table 4-1 — EBNF Operators	3
	Table 5-1 — Sensor URN [identifier-type] values	5
	Table 5-2 — Sensor URN [identifier-type-dependent] values	5

Table 5-3 — <SensorEvents> event= attribute allowed values.....	7
Table 5-4 — State Variables for Eventing	8
Table 5-5 — Actions	9
Table A.1 — IoT Management and Control General Parameters.....	11
Table A.2 — SensorEventEnable parameter allowed values.....	15
Table A.3 — Sensor Permissions	17
Table A.4 — GroupSetType values.....	19
Table A.5 — ScriptSettings Parameters for predefined Settings	20
Table A.6 — Default permission settings	21
Table A.7 — Default script permission settings	22
Table A.8 — Default permissions settings.....	23
Table A.9 —ScriptSettings Parameters for Scripts	24
Table A.10 — Script Attributes	25
Table A.11 — Script Status values	26
Table A.12 — Script default permissions	27
Table A.13 — Property-identifier values for Scripts	28
Table A.14 — property-identifier values for collections.....	29
Table A.15 — Sample Collection for Examples	31
Table A.16 — Sample Collection for Living Room Lights	35
Table A.17 — Continue Collection for Living Room Lights.....	36
Table A.18 — Collection for Living Room Lights - Timer example.....	37
Table B.1 — IoT Management and Control required Dataltem(s).....	38
Table C.1 — Common Device Identifiers	39
Table C.2 — mapping of mandatory Dataltems per Common Device Identifier	43
Table D.1 — IEEE-11073 specific Medical Device System Parameters.....	50
Table D.2 — IEEE-11073 Medical Device Sensor Parameters.....	53
Table D.3 — Default Dataltem(s) for IEEE-11073 Medical Object Class Sensors	54
Table D.4 — Dataltem(s) for Medical Object Class Attributes	55
Table D.5 — Dataltem(s) for Medical Object Class Observations	56
Table D.6 — IEEE-11073 Persistent Metric Store Parameter Nodes	57
Table D.7 — IEEE-11073 Persistent Metric Segment Parameter Nodes.....	59
Table D.8 — IEEE-11073 Persistent Metric Segment Dataltem(s)	60

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](http://www.iso.org/iso/8601).
- [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>.

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/smgmt/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/smgmt/sdmevent-v1-20130701.xsd>. Latest version available at: <http://www.upnp.org/schemas/smgmt/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/smgmt/sdmdid-v1-20130701.xsd>. Latest version available at: <http://www.upnp.org/schemas/smgmt/sdmdid.xsd>.

[19] XML Schema UPnP IoT Management and Control DataRecord Information, UPnP Forum, July 1, 2013. Available at: <http://www.upnp.org/schemas/smgmt/srecinfo-v1-20130701.xsd>. Latest version available at: <http://www.upnp.org/schemas/smgmt/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
<code>::=</code>	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.
<code> </code>	alternative separator – separates sequences on the right that are independently allowed definitions for the non-terminal on the left.
<code>*</code>	null repetition – means the expression to its left may occur zero or more times.
<code>+</code>	non-null repetition – means the expression to its left shall occur at least once and may occur more times.
<code>[]</code>	optional – the expression between the brackets is allowed.
<code>()</code>	grouping – groups the expressions between the parentheses.
<code>-</code>	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 Dataltem(s). Sensor Dataltem(s) are described by providing a name, format, encoding (see UPnP IoT Management and Control Architecture Overview [10], subclause 4.3, "Dataltem 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 Dataltem(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 Dataltem(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 Dataltems

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

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	<u>smgt-sct</u>
Sensor Type	<u>smgt-st</u>
Sensor URN	<u>smgt-surn</u>

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
<u>monitor</u>	<u>smgt-st</u> , <u>smgt-surn</u>	Dataltem(s) or Sensor(s) intended to periodically monitor or record device performance over a period of time.
<u>status</u>	<u>smgt-st</u> , <u>smgt-surn</u>	Dataltem(s) or Sensor(s) intended to provide current device status or settings.
<u>datafeed</u>	<u>smgt-st</u> , <u>smgt-surn</u>	Dataltem(s) or Sensor(s) which to provide a continuous source of sensor data.
<u>telemetry</u>	<u>smgt-st</u> , <u>smgt-surn</u>	Dataltem(s) or Sensor(s) intended to report data obtained by a Sensor.
<u>setting</u>	<u>smgt-st</u> , <u>smgt-surn</u>	Dataltem(s) or Sensor(s) intended to modify end-user settable characteristics of a device.
<u>control</u>	<u>smgt-st</u> , <u>smgt-surn</u>	Dataltem(s) or Sensor(s) intended to control aspects of a device not typically set by end-users
<u>alarms</u>	<u>smgt-st</u> , <u>smgt-surn</u>	Dataltem(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 |
|---|--|
| <u>SensorCollectionAdded</u> | A new sensor collection has been added.
Reporting of this event is enabled by default. |
| <u>SensorCollectionRemoved</u> | A previously reported sensor collection has been removed.
Reporting of this event is enabled by default. |
| <u>SensorAdded</u> | A new sensor has been added.
Reporting of this event is enabled by default. |
| <u>SensorRemoved</u> | A previously reported sensor has been removed.
Reporting of this event is enabled by default. |
| <u>SOAPDataAvailable</u> | The sensor has data available to be read by the <u>ReadSensor()</u> SOAP action.
Reporting of this event is disabled by default. Reporting of this event can be enabled by setting the <u>SensorEventsControl</u> parameter to "SOAPDataAvailableEnable,1". |
| <u>SOAPDataOverrun</u> | The sensor has discarded data which was not read by any SOAP client.
Reporting of this event is disabled by default. Reporting of this event can be enabled by setting the <u>SensorEventControl</u> parameter to "SOAPDataOverrunEnable,1". |
| <u>TransportDataAvailable</u> | The sensor has received data available to be sent to one or more transport connection for this sensor.
Reporting of this event is disabled by default. Reporting of this event can be enabled by setting the <u>SensorEventControl</u> parameter to "TransportDataAvailableEnable,1". |
| <u>TransportDataOverrun</u> | The sensor has discarded data which was not read by any active transport connection.
Reporting of this event is disabled by default. Reporting of this event can be enabled by setting the <u>SensorEventControl</u> parameter to "TransportDataOverrunEnable,1". |
| <u>TransportConnectionError</u> | An error has occurred on one or more active transport connections for this sensor.
Reporting of this event is disabled by default. Reporting of this event can be enabled by setting the <u>SensorEventControl</u> parameter to "TransportConnectionErrorEnable,1". |
| <u>SensorAvailable</u> | A currently configured sensor has become available.
Reporting of this event is disabled by default. Reporting of this event can be enabled by setting the <u>SensorEventControl</u> parameter to "SensorAvailabilityEnable,1". |
| <u>SensorUnavailable</u> | 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 <u>SensorEventControl</u> 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
R/A ^a | IOTManagement and Control
R/A ^a |
|---|-------------------------|---|
| <i>ConfigurationUpdate</i> | <i>R</i> | <i>R</i> |
| <i>CurrentConfigurationVersion</i> | <i>R</i> | <i>R</i> |
| <i>SupportedDataModelsUpdate</i> | <i>R</i> | <i>R</i> |
| <i>SupportedParametersUpdate</i> | <i>R</i> | <i>R</i> |
| <i>AttributeValuesUpdate</i> | <i>A</i> | <i>A</i> |
| <i>InconsistentStatus</i> | <i>A</i> | <i>A</i> |
| <i>AlarmsEnabled</i> | <i>CR</i> | <i>R</i> |
| ^a <i>R</i> = required, <i>A</i> = allowed, <i>CR</i> = conditionally required, <i>CA</i> = conditionally allowed, <i>X</i> = Non-standard, add <i>-D</i> when deprecated (e.g., <i>R-D</i> , <i>A-D</i>). | | |

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
R/A ^a | IoT Management and
Control
R/A ^b | IoT Management and
Control CP
R/A ^c |
|--|-------------------------|---|--|
| <u>GetSupportedDataModels()</u> | <u>R</u> | <u>R</u> | <u>R</u> |
| <u>GetSupportedParameters()</u> | <u>R</u> | <u>R</u> | <u>R</u> |
| <u>GetInstances()</u> | <u>R</u> | <u>R</u> | <u>R</u> |
| <u>GetValues()</u> | <u>R</u> | <u>R</u> | <u>R</u> |
| <u>GetSelectedValues()</u> | <u>A</u> | <u>A</u> | <u>A</u> |
| <u>SetValues()</u> | <u>A</u> | <u>R</u> | <u>A</u> |
| <u>CreateInstance()</u> | <u>A</u> | <u>CR</u> | <u>A</u> |
| <u>DeleteInstance()</u> | <u>A</u> | <u>CR</u> | <u>A</u> |
| <u>GetAttributes()</u> | <u>R</u> | <u>R</u> | <u>A</u> |
| <u>SetAttributes()</u> | <u>A</u> | <u>A</u> | <u>A</u> |
| <u>GetInconsistentStatus()</u> | <u>A</u> | <u>A</u> | <u>A</u> |
| <u>GetConfigurationUpdate()</u> | <u>R</u> | <u>R</u> | <u>A</u> |
| <u>GetCurrentConfigurationVersion()</u> | <u>R</u> | <u>R</u> | <u>A</u> |
| <u>GetSupportedDataModelsUpdate()</u> | <u>R</u> | <u>R</u> | <u>A</u> |
| <u>GetSupportedParametersUpdate()</u> | <u>R</u> | <u>R</u> | <u>A</u> |
| <u>GetAttributeValuesUpdate()</u> | <u>A</u> | <u>A</u> | <u>A</u> |
| <p>^a For a device this column indicates whether the action shall be implemented or not, where <u>R</u> = required, <u>A</u> = allowed, <u>CR</u> = conditionally required, <u>CA</u> = conditionally allowed, <u>X</u> = Non-standard, add <u>-D</u> when deprecated (e.g., <u>R-D</u>, <u>A-D</u>).</p> <p>^b For a device this column indicates whether the action shall be implemented or not, where <u>R</u> = required, <u>A</u> = allowed, <u>CR</u> = conditionally required, <u>CA</u> = conditionally allowed, <u>X</u> = Non-standard, add <u>-D</u> when deprecated (e.g., <u>R-D</u>, <u>A-D</u>).</p> <p>^c For a control point this column indicates whether a control point shall be capable of invoking this action, where <u>R</u> = required, <u>A</u> = allowed, <u>CR</u> = conditionally required, <u>CA</u> = conditionally allowed, <u>X</u> = Non-standard, add <u>-D</u> when deprecated (e.g., <u>R-D</u>, <u>A-D</u>).</p> | | | |

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 MultilInstance 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 MultilInstance 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/#/ | MultInstance | 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/#/ | MultInstance | 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/#/ | MultInstance | RO | CR | A.1.1.45 | | 0 |
| MemberID (CollectionID or SensorID) | string | RW | R | A.1.1.46 | | 0 |
| | | | | | | |
| /UPnP/SensorMgt/SensorCollections/#/ | MultInstance | RO | R | A.1.1.4 | | 0 |
| ParticipatingGroupSetID | string | RW | CR | A.1.1.47 | | 0 |
| | | | | | | |
| /UPnP/SensorMgt/SensorCollections/#/Sensors/#/ | MultInstance | 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/](#) MultInstance 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/](#) MultInstance 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](#) MultInstance node entries.

A.1.1.4 /UPnP/SensorMgt/SensorCollections/**Type:** MultInstance

Description: The required [SensorCollections](#) MultInstance 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 [SensorsNumberOfEntries](#) parameter shall provide the number of [Sensor](#) MultInstance node entries.

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

Type: MultInstance

Description: The required [Sensors](#) MultInstance 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/1/Sensors/1/SensorsRelatedNumberOfEntries

Type: int

Description: The conditionally required [SensorsRelatedNumberOfEntries](#) parameter shall provide the number of [SensorsRelated](#) MultInstance node entries. This parameter shall be implemented if the [SensorsRelated](#) MultInstance node is supported.

A.1.1.20 /UPnP/SensorMgt/SensorCollections/1/Sensors/1/SensorGroupsNumberOfEntries

Type: int

Description: The conditionally required [SensorsGroupsOfEntries](#) parameter shall provide the number of [SensorGroups](#) MultInstance node entries. This parameter shall be implemented if the [SensorGroups](#) MultInstance node is supported.

A.1.1.21 /UPnP/SensorMgt/SensorCollections/1/Sensors/1/SensorDefaultPermissionsNumberOfEntries

Type: int

Description: The conditionally required [SensorsPermissionNumberOfEntries](#) parameter shall provide the number of [SensorPermissions](#) MultInstance node entries. . This parameter shall be implemented if the [SensorDefaultPermissions](#) MultInstance node is supported

A.1.1.22 /UPnP/SensorMgt/SensorCollections/1/Sensors/1/SensorsRelated/

Type: MultInstance

Description: The allowed [SensorsRelated](#) MultInstance 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/1/Sensors/1/SensorsRelated/1/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](#) MultInstance node and shall end at the [Sensors/1](#) instance node identifying the sensor being referenced.

A.1.1.24 /UPnP/SensorMgt/SensorCollections/1/Sensors/1/SensorGroups/

Type: MultInstance

Description: The conditionally required [SensorGroups](#) MultInstance node includes zero or more [SensorGroup](#) parameter nodes identifying SensorGroup(s) this sensor is part of.

The **SensorGroups** MultilInstance 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: MultilInstance

Description: The conditionally allowed **SensorPermissions** MultilInstance 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](#) MultInstance node entries.

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

Type: MultInstance

Description: The required [SensorURNs](#) MultInstance node lists URN values for sets of DatalItem(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 DatalItem(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/DatalItemsNumberOfEntries

Type: int

Description: The required [DatalItemsNumberOfEntries](#) parameter shall provide the number of [DatalItems](#) MultInstance node entries.

A.1.1.34 /UPnP/SensorMgt/SensorCollections/#!/Sensors/#!/SensorURNs/#!/DatalItems/

Type: MultInstance

Description: The required [DatalItems](#) MultInstance node identifies sets of DatalItem(s) supported for the corresponding [SensorURN](#) parameter value.

A.1.1.35 /UPnP/SensorMgt/SensorCollections/#!/Sensors/#!/SensorURNs/#!/DatalItems/#!/Name

Type: MultInstance

Description: The required [Name](#) parameter identifies the name of an individual DatalItem. See IoT Management and Control Architecture Overview [10] subclause 4.3, "DatalItem semantics" for further information.

A.1.1.36 /UPnP/SensorMgt/SensorCollections/#!/Sensors/#!/SensorURNs/#!/DatalItems/#!/Type

Type: MultInstance

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

A.1.1.37 /UPnP/SensorMgt/SensorCollections/#!/Sensors/#!/SensorURNs/#!/DatalItems/#!/Encoding

Type: MultInstance

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

A.1.1.38 /UPnP/SensorMgt/SensorCollections/#!/Sensors/#!/SensorURNs/#!/DatalItems/#!/Description

Type: MultInstance

Description: The required [Description](#) parameter shall provide a string containing an XML document which describes the named DatalItem or an empty string if no DatalItem Description document is available. This XML document shall comply with the DatalItem 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 |
|--|--|
| sgmt-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. |
| sgmt-gst:control-group | GroupSet intended to define a control group in which members will be controlled as a single entity.
No GroupSetSpecific instance is required if all members have the same control scheme defined in the SensorURNs .
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.
If no GroupSetSpecific instance is defined, the common set of SensorURNs defined between the members will be used for control. |
| sgmt-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 predefined *DataRecords* to be applied to a group of sensors. For example, it is possible to predefined 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 allowed 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](#) MultilInstance node is supported**A.2.1.16 /UPnP/ScriptSettings/Settings/#!/SettingPermissions/#!/****Type:** MultilInstance**Description:** The conditionally allowed [SettingPermissions](#) MultilInstance 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 . |