INTERNATIONAL STANDARD

ISO/IEC 29341-30-12

First edition
2017-06

Information technology — UPnP Device Architecture —

Part 30-12: IoT management and control device control protocol — IoT management and control transport generic service

Technologies de l'information — Architecture de dispositif UPnP —

Partie 30-12: Protocole de contrôle de dispositif de gestion et de contrôle de l'Internet des objets — Service de transport générique de gestion et de contrôle de l'Internet des objets

Reference number
ISO/IEC 29341-30-12:2017(E)

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 1
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
5 Service Modelling Definitions 3
5.1 Service Type 3
5.2 SensorTransportGeneric Service Architecture 3
5.3 Key Concepts 3
5.3.1 Sensor Connectivity 3
5.3.2 Sensor HTTP/HTTPS Transport Protocol 4
5.4 State Variables 5
5.4.1 State Variable Overview 5
5.4.2 *A_ARG_TYPE_SensorID* 5
5.4.3 *A_ARG_TYPE_SensorURN* 5
5.4.4 *A_ARG_TYPE_SensorClientID* 5
5.4.5 *A_ARG_TYPE_DataRecords* 5
5.4.6 *A_ARG_TYPE_DataRecordCount* 7
5.4.7 *A_ARG_TYPE_TransportURL* 7
5.4.8 *A_ARG_TYPE_SensorDataTypeEnable* 7
5.4.9 *A_ARG_TYPE_SensorRecordInfo* 7
5.4.10 *A_ARG_TYPE_TransportConnectionID* 8
5.4.11 *A_ARG_TYPE_TransportConnections* 8
5.5 Actions 9
5.5.1 *ConnectSensor()* 9
5.5.2 *DisconnectSensor()* 12
5.5.3 *ReadSensor()* 13
5.5.4 *WriteSensor()* 14
5.5.5 *GetSensorTransportConnections()* 15
5.5.6 Error Code Summary 16
6 XML Service Description 17

Table 1 — State Variables 5
Table 2 — Actions 9
Table 3 — Arguments for *ConnectSensor()* 9
Table 4 — Error Codes for *ConnectSensor()* 11
Table 5 — Arguments for *DisconnectSensor()* 12
Table 6 — Error Codes for *DisconnectSensor()* 12
Table 7 — Arguments for *ReadSensor()* 13
Table 8 — Error Codes for *ReadSensor()* 14
Table 9 — Arguments for *WriteSensor()* 14

Table 10 — Error Codes for *WriteSensor()* 15
Table 11 — Arguments for *GetSensorTransportConnections()* 16
Table 12 — Error Codes for *GetSensorTransportConnections()* 16
Table 13 — Error Code Summary.....16

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 29341-30-12 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 the characteristics of a networked service that offers *sensor* and/or *actuator* data transport capabilities to an independent networked entity known as a control point.

This document defines the service SensorTransportGeneric:1 (in short TransportGeneric), which identifies Level *1* of the UPnP IoT Management and Control TransportGeneric:1 Service [10]. This Publicly Available Specification is applicable to Standardized DCPs of the UPnP Forum which include this service.

This service enables UPnP clients to access sensors and/or actuators without needing a detailed knowledge of the target sensor or actuator or its connectivity to the UPnP network. *Sensors* and *Actuators* are instead treated a generic data sources or sinks.

This service definition is compliant with the UPnP Device Architecture version 1.0 [1]. It defines a service type referred to herein as TransportGeneric service.

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] 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.

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

[10] UPnP UPnP IoT Management and Control Device TransportGeneric:1 Service, UPnP Forum July 1, 2013. Available at: http://www.upnp.org/specs/iotmc/UPnP-iotmc-UPnP

IoTManagementandControl-TransportGeneric-v1-Service-20130701.pdf. Latest version available at: http://www.upnp.org/specs/iotmc/UPnP-iotm-IoTManagementandControl-TransportGeneric-v1-Service.pdf.

[11] 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/smgt/UPnP-ds-DataStore-v1-Service.pdf.

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

[13] 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.

[14] UPnP ConfigurationManagement:2 Service, UPnP Forum, February 16, 2012. 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.

[15] XML Schema UPnP DataStore DataRecord Status, UPnP Forum, July 1, 2013. Available at: http://www.upnp.org/schemas/ds/drecstatus-v1-20130701.xsd. Latest version available at: http://www.upnp.org/schemas/ds/drecstatus.xsd.

[16] XML Schema UPnP DataStore DataRecord, UPnP Forum, July 1, 2013. Available at: http://www.upnp.org/schemas/ds/drecs-v1-20130701.xsd. Latest version available at: http://www.upnp.org/schemas/ds/drecs.xsd.

[17] XML Schema for Sensor 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.

[18] XML Schema for Sensor TransportConnections Information, UPnP Forum, July 1, 2013. Available at: http://www.upnp.org/schemas/smgt/tspc-v1-20130701.xsd.
Latest version available at: http://www.upnp.org/schemas/smgt/tspc.xsd.

3 Terms, Definitions and Abbreviations

For the purposes of this document, the terms and definitions given in [1] and [8] 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 UPnP Device Architecture, version 1.0 [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 UPnP Device Architecture, version 1.0 [1], Clause 2.5, "Description: Non-standard vendor extensions".

5 Service Modelling Definitions

5.1 Service Type

The following URN identifies a service that is compliant with this specification:

urn:schemas-upnp-org:service:SensorTransportGeneric:1

SensorTransportGeneric or TransportGeneric service is used herein to refer to this service type.

5.2 SensorTransportGeneric Service Architecture

The TransportGeneric service enables UPnP clients to obtain sensor data without needing to have detailed understanding the operation of a target sensor or the sensor's access network protocols. This service abstracts these notions treating the sensor as a generic data source which defines output record formats. Both HTTP transport and a SOAP-based read action are defined. In the case of actuators, this service treats the target as a generic data sink and defines a SOAP-based write action. This service defines sections of the IoT Management And Control DataModel related to the description of sensors supported by this service. See UPnP IoT Management and Control DataModel service [12] and UPnP IoT Management and Control Architecture Overview [8] for additional details.

5.3 Key Concepts

Note: See IoT Management and Control Architecture Overview [8] for an overall discussion of IoT Management and Control and DataStore services.

5.3.1 Sensor Connectivity

The TransportGeneric service specification defines a set of SOAP actions which manage connectivity to sensors as generic data sources and sinks. Sensors are advertised to UPnP home-network clients via the IoTM anagement And Control DataModel service [12]. The DataModel parameter *SensorID* is used to identify a sensor to SOAP actions defined by this service.

Two models are supported for obtaining data from Sensors.

1) HTTP/HTTPS Transport Model

   The TransportGeneric service acts as an HTTP/HTTPS client. The URL of a HTTP/HTTPS transport sever is provided to this service using the *ConnectSensor()* action. When the Sensor has data available the TransportGeneric service connects to

the transport server endpoint provided and posts one or more DataRecords to that endpoint.

2) SOAP Action Model

The TransportGeneric service provides a *ReadSensor()* action. A UPnP control point can issue this action in response to a Sensor data event to read one or more pending DataRecords from the target sensor.

The TransportGeneric service shall support both of these models enabling a single SOAP client and one or more HTTP/HTTPS transport connections to exist concurrently. Each connection maintains its state independently, so that read activity on a particular connection has no effect on the data available to other connections.

For providing data to Sensors the *WriteSensor()* action is provided.

Note: Future versions of the UPnP IoT Management and Control device are anticipated to define additional transport-specific (SensorTransport*) services to provide access to low-level control of bridged sensor transport networks.

5.3.2 Sensor HTTP/HTTPS Transport Protocol

Sensor connections initialized by the *ConnectSensor()* action shall support the HTTP/HTTPS Transport model as follows:

1) When the managed sensor has data available, the TransportGeneric service shall issue an HTTP/HTTPS POST request to the transport endpoint provided by the TransportURL argument of the *ConnectSensor()* action.
2) The request shall contain any HTTP entity-headers as required by RFC-2616.
3) The entity-body shall contain a DataRecords XML document as described for the *A_ARG_TYPE_DataRecords* state variable.
4) If all <datarecord> elements contained within the POST request are acceptable, then the transport endpoint shall generate a HTTP-response with HTTP status 200 and an empty entity-body.
5) If transport endpoint does not accept all of the POST(ed) <datarecord> elements, the transport endpoint shall return a HTTP-response with HTTP status 200 with an entity-body containing an XML document conforming to the XML Schema UPnP DataStore DataRecord Status [15].

Note: This schema is shared with the DataStore service [11].

```
<?xml version="1.0" encoding="utf-8"?>
<DataRecordsStatus
   xmlns="urn:schemas-upnp-org:ds:drecstatus"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:schemas-upnp-org:ds:drecstatus
      http://www.upnp.org/schemas/ds/drecstatus-v1.xsd">

   <!-- For each <datarecord> element in the POST request -->
   <datarecordstatus accepted="0|1" />

    ... Additional <datarecordstatus> elements ...

</DataRecordsStatus>
```

<?xml>

Required. Case Sensitive.

<DataRecordsStatus>

Required. Shall include the namespace declaration for the XML Schema UPnP DataStore DataRecord Status (urn:schema-upnp-org:ds:drecstatus). Shall include the following elements and attributes:

<datarecordstatus>

Required. XML. For each <datarecord> element in the original HTTP-POST request, a corresponding <datarecordstatus> element shall be included.

`accepted`

Required. Boolean. This attribute shall be set to “1” if the corresponding `<datarecord>` element was accepted by the transport server and to “0” if the corresponding `<datarecord>` was rejected.

5.4 State Variables.

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.

5.4.1 State Variable Overview

Table 1 — State Variables

Variable Name	R/A [a]	Data Type	Reference
A_ARG_TYPE_SensorID	*R*	**string**	See 5.4.2
A_ARG_TYPE_SensorURN	*R*	**string**	See 5.4.3
A_ARG_TYPE_SensorClientID	*R*	**string**	See 5.4.4
A_ARG_TYPE_DataRecords	*CR* [b]	**string**(XML fragment)	See 5.4.5
A_ARG_TYPE_DataRecordCount	*R*	**ui4**	See 5.4.6
A_ARG_TYPE_TransportURL	*R*	**string**	See 5.4.7
A_ARG_TYPE_SensorDataTypeEnable	*R*	**boolean**	See 5.4.8
A_ARG_TYPE_SensorRecordInfo	*R*	**string**	See 5.4.9
A_ARG_TYPE_TransportConnectionID	*R*	**string**	See 5.4.10
A_ARG_TYPE_TransportConnections	*R*	**string**	See 5.4.11

[a] *R* = REQUIRED, *A* = ALLOWED, *CR* = CONDITIONALLY REQUIRED, *CA* = CONDITIONALLY ALLOWED, *X* = Non-standard, add *-D* when deprecated (e.g., *R-D*, *O-D*).

[b] This conditionally required state variable MUST be implemented if actions *ReadSensor()* or *WriteSensor()* are implemented.

5.4.2 *A_ARG_TYPE_SensorID*

This state variable contains a **string** which shall indicate a sensor that is managed by this IoT Management and Control device.

5.4.3 *A_ARG_TYPE_SensorURN*

This state variable contains a **string** which shall identify a URN value for a target sensor. The URN value is used to match to a set of DataItems the target sensor can provide. See UPnP IoT Management and Control DataModel service [12] *SensorURN* for further detail on sensor URN values.

5.4.4 *A_ARG_TYPE_SensorClientID*

This state variable contains a **string** which the target sensor shall provide as the value of the sensor's DataItem named "ClientID".

5.4.5 *A_ARG_TYPE_DataRecords*

This state variable contains an XML document which conforms to the XML Schema UPnP DataStore DataRecord [16]. This document is used to retrieve or convey DataRecord(s) to or from the *TransportGeneric* service.

Note: This schema is shared with the DataStore service [11].

```
<?xml version="1.0" encoding="UTF-8"?>
<DataRecords xmlns="urn:schemas-upnp-org:ds:drecs"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:schemas-upnp-org:ds:drecs
      http://www.upnp.org/schemas/ds/drecs-v1.xsd">

   <datarecord>
      <field    name="Name of DataItem"
                type="Type of DataItem"
                encoding="Encoding of DataItem"
                namespace="namespace(For XML-based DataItem)">
         Value of DataItem
      </field>

      ...Additional <field> elements for DataItems in this record...

   </datarecord>

   ...Additional <datarecord> elements ...

</DataRecords>
```

`<?xml>`

Allowed. Case sensitive.

`<DataRecords>`

Required. The `datarecords` element shall contain one or more `<datarecord>` child elements; each `datarecord` element providing the contents of an individual DataRecord. Shall include the following elements. Shall include a namespace declaration for the XML Schema UPnP DataStore DataRecord ("urn:schemas-upnp-org:ds:drecs").

`<datarecord>`

Required. xsd:string. Each `<datarecord>` element shall contain zero or more `<field>` elements.

`<field>`

Required. xsd:string. Each `<datarecord>` element shall contain zero or more `<field>` elements. The value of this element carries the value of the corresponding named DataItem as indicated by the field element's name attribute.

`name`

Required. xsd:string. Each `<field>` element shall designate a corresponding DataItem by specifying its name as the value this attribute. A `<datarecord>` element is prohibited from containing multiple `<field>` elements with identical `name` attribute values.

`type`

Allowed. xsd:string. This attribute shall provide data type information for each DataItem within a DataRecord. See IoTManagementAndControl Architecture Overview [8] subclause 4.3, "DataItem Semantics" for encoding of the type attribute.

Note: DataItem type information is conveyed when transport connections are established rather than as part of each DataRecord. This information is provided for testing and diagnostic purposes.

`encoding`

Required. xsd:string. . This attribute shall provide the encoding for this DataItem. Allowable values for this attribute are either "*ascii*", "*utf-8*" or "*base64*".

Note: DataItem encoding information is conveyed when transport connections are established rather than as part of each DataRecord. This information is provided for testing and diagnostic purposes.

`namespace`

Allowed. xsd:string. This attribute is permitted for DataItems consisting of strings containing XML compliant documents. This attribute shall provide the expected namespace for the encoded document. If this attribute is present, implementations shall validate that the corresponding DataItem is a valid

XML document encoded following XML escaping rules. Implementations are permitted to perform further validate to insure consistency of the DataItem with the XML namespace indicated by this attribute

Note: DataItem namespace information is conveyed when transport connections are established rather than as part of each DataRecord. This information is provided for testing and diagnostic purposes.

5.4.6 *A_ARG_TYPE_DataRecordCount*

This state variable contains an unsigned integer (**ui4**) which shall indicate a count of *DataRecord(s)*.

5.4.7 *A_ARG_TYPE_TransportURL*

This state variable shall define a **string** argument that conforms to Uniform Resource Locator syntax [4]. The purpose of this URL is to enable IoT Management and Control device to submit asynchronous updates to transport endpoint identified by this URL. This URL may be allocated by DataStore service [11].

5.4.8 *A_ARG_TYPE_SensorDataTypeEnable*

This state variable shall provide a **boolean** value. If this argument is set to "1" (one) then each DataRecord <field> element provided by the subject action shall include type, encoding and namespace (if applicable) attributes for the corresponding DataItem.

5.4.9 *A_ARG_TYPE_SensorRecordInfo*

This state variable contains an XML document which conforms to the Sensor DataRecord Information schema [17]. This XML document which defines the contents of DataRecords to be transmitted by the SensorTransportGeneric service either via a *ReadSensor()* / *WriteSensor()* SOAP actions or via a sensor transport connection established via the *ConnectSensor()* action.

```
<?xml version="1.0" encoding="utf-8"?>
<SensorRecordInfo
   xmlns="urn:schemas-upnp-org:smgt:srecinfo"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:schemas-upnp-org:smgt:srecinfo
      http://www.upnp.org/schemas/smgt/srecinfo-v1.xsd"
   >
   <sensorrecord>
      <field
         name="Name of DataItem"
         prefix="Prefix to be applied to name of DataItem"
      </field>

      ... Additional <field> element(s) for DataItems to be included in this
      ... Sensor DataRecord ...

   </sensorrecord>

</SensorRecordInfo>
```

<?xml>

Allowed. Case sensitive.

<SensorRecordInfo>

Required. Shall include a namespace declaration for the XML Schema for Sensor DataRecord Information ("urn:schemas-upnp-org:smgt:srecinfo"). Shall include the following elements and attributes:

<sensorrecord>

Allowed. Shall be specified one time. Shall include the following elements and attributes:

`<field>`

Allowed. Shall be specified zero or more times. Each `<field>` element shall specify a DataItem name supported by the target sensor to be included in `<datarecord>` element(s) generated by the sensor.

`name`

Required. Shall provide the name of the DataItem to be included in the `<datarecord>` generated by the target sensor.

`prefix`

Allowed. The value provided shall be prefixed to the DataItem name in the `<datarecord>` generated by the target sensor.

5.4.10 *A_ARG_TYPE_TransportConnectionID*

This state variable shall contain a unique identifier for a sensor transport connection. See action(s) *ConnectSensor()* and *DisconnectSensor()* for usage of this argument type.

5.4.11 *A_ARG_TYPE_TransportConnections*

This state variable contains an XML document which conforms to the XML Schema UPnP TransportConnections Information [18].

```
<?xml version="1.0" encoding="UTF-8"?>
<TransportConnections xmlns="urn:schemas-upnp-org:smgt:tspc"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:schemas-upnp-org:smgt:tspc
      http://www.upnp.org/schemas/smgt/tspc-v1.xsd">

   <transportconnection
         sensorID="SensorID value for the reported sensor"
         transportConnectionID="identifier for the transport connection"
         transportURL="transport endpoint URL for the transport connection"
         sensorClientID="ClientID argument from the ConnectSensor() action" />

      ...Additional <transportconnection> elements for each transport connections...

   </transportconnection>

</TransportConnections>
```

`<?xml>`

Allowed. Case sensitive.

`<TransportConnections>`

Required. XML. Shall include a namespace declaration for the XML Schema UPnP TransportConnections (“urn:schemas-upnp-org:smgt:tspc”). Shall include zero or more of the following elements:

`<transportconnection>`

Required. xsd:string. Each `<transportconnection>` element provides information for a transport connection supported by the sensor indicated by the `sensorID` attribute. This element shall contain the following attributes.

`sensorID`

Required. xsd:string. The value of this attribute shall identify the sensor supporting the transport connection described by this `<transportconnection>` element. This value corresponds to the *SensorID* argument value supplied to the *ConnectSensor()* action.

`transportConnectionID`

Required. xsd:string. The value of this attribute shall identify the transport connection described by this `<transportconnection>` element. This value corresponds to the *TransportConnectionID* argument value returned by the *ConnectSensor()* action.

`transporURL`

Required. xsd:string. The value of this attribute shall identify the transport endpoint URL for the transport connection described by this `<transportconnection>` element. This value corresponds to the *TransportURL* argument value supplied to the *ConnectSensor()* action.

`sensorClientID`

Required. xsd:string. The value of this attribute shall provide an identifier provided by the control point assigned to the transport connection described by this `<transportconnection>` element. The value corresponds to the *SensorClientID* argument value supplied to the *ConnectSensor()* action.

5.5 Actions

Table 2 — Actions

Name	Device R/A [a]	Control Point R/A [b]
ConnectSensor()	R	R
DisconnectSensor()	R	R
ReadSensor()	A	A
WriteSensor()	A	A
GetSensorTransportConnections	R	A

[a] For a device this column indicates whether the action MUST 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*, *O-D*).

[b] For a control point this column indicates whether a control point MUST 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*, *O-D*).

5.5.1 *ConnectSensor()*

This required action shall connect a *sensor* (*SensorID*) to a *transport endpoint* identified by the *TransportURL* argument. The *SensorTransportGeneric* service may connect or disconnect to the URL identified as needed to transfer data sensor data to the *transport endpoint*. Data transferred by this service shall consist of `<DataRecord>` elements (see the XML Schema UPnP DataStore DataRecord [16]). The *SensorTransportGeneric* service shall support at least one transport connection per unique *SensorID* value. A sensor supported by this service should allow additional transport connections to unique transport endpoints.

5.5.1.1 Arguments

Table 3 — Arguments for *ConnectSensor()*

Argument	Direction	relatedStateVariable
SensorID	*IN*	*A_ARG_TYPE_SensorID*
SensorClientID	*IN*	*A_ARG_TYPE_SensorClientID*
SensorURN	*IN*	*A_ARG_TYPE_SensorURN*
SensorRecordInfo	*IN*	*A_ARG_TYPE_SensorRecordInfo*
SensorDataTypeEnable	*IN*	*A_ARG_TYPE_SensorDataTypeEnable*
TransportURL	*IN*	*A_ARG_TYPE_TransportURL*
TransportConnectionID	*OUT*	*A_ARG_TYPE_TransportConnectionID*

5.5.1.2 Argument Descriptions

SensorID: The value of this string argument shall identify a (target) sensor which is managed by IoT Management and Control device. See the *SensorID* parameter in IoT Management and Control DataModel service [12] for further details about identifying sensors and their properties.

SensorClientID: The value of this string argument shall be returned as the value of the *DataItem* named *ClientID* by the target sensor. See IoT Management and Control DataModel service Annex B "Required Sensor DataItems" for additional information.

SensorURN: This argument shall provide shall identify a URN value for a target sensor. The URN value is used to match to a set of DataItems the target sensor can provide. See UPnP IoT Management and Control DataModel service [12] *SensorURN* for further detail on sensor URN values.

SensorRecordInfo: This argument shall provide a string containing an XML document which identifies the contents of the DataRecord(s) to be provided by the target sensor. See *A_ARG_TYPE_SensorRecordInfo* for further details.

SensorDataTypeEnable: If this argument is set to "1" (one) then each DataRecord <field> element shall include type, encoding and namespace (if applicable) attributes for the corresponding DataItem.

Note: Inclusion of DataItem type information is primarily for testing and diagnostic purposes to allow the transport client to verify the expected DataItem types were provided.

TransportURL: This argument shall provide a URL to a Sensor Transport server. The URL provided shall accept DataRecord(s) as described by the *SensorRecordInfo* argument transmitted via HTTP(S) protocols per RFC-2616 [4]. See subclause 5.3.2, "Sensor HTTP/HTTPS Transport Protocols" for details of the Sensor DataRecord Transport protocols.

TransportConnectionID: This argument shall return a unique identifier for the transport connection.

5.5.1.3 Service Requirements

If this IoT Management and Control device implements *DeviceProtection*, then a Control Point identity invoking this action shall have either a *smgt:ConnectSensor* or *Admin* role for the target sensor to successfully complete this action. See the *SensorPermissions* parameter as described in UPnP IoT Management and Control DataModel service [12] for further details on the sensor permission model.

Implementation(s) should generate unique *TransportConnectionID* values across service restarts.

5.5.1.4 Control Point Requirements When Calling The Action

None.

5.5.1.5 Dependency on Device State

A IoT Management and Control implementation may set implementation limits on the number of connections allowed to a target sensor. However, an implementation shall support at least one transport connection per sensor and should support additional transport connections.

5.5.1.6 Effect on Device State

The IoT Management and Control device shall retain the provided *TransportURL*. The IoT Management and Control device may connect to the *TransportURL* to send DataRecord(s) to the transport server endpoint at any time. The IoT Management and Control device should retry failed attempts to contact the transport server endpoint.

5.5.1.7 Errors

Table 4 — Error Codes for *ConnectSensor()*

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture clause on Control.
500-599	TBD	See UPnP Device Architecture clause on Control.
600-699	TBD	See UPnP Device Architecture clause on Control.
701	Invalid XML argument	An argument value (SensorRecordInfo) provided does not comply with the corresponding schema requirements.
702	SensorID not found	The SensorID provided does not correspond to a known sensor.
703	Sensor URN not found	The SensorURN provided does not correspond to a known URN for the indicated SensorID.
705	Sensor DataItem not found	A DataItem referenced by the SensorRecordInfo XML argument cannot be found.
708	Transport connection limit exceeded	The number of available transport connections to the indicated SensorID has been exceeded.

5.5.2 *DisconnectSensor()*

This required action shall disconnect a sensor from the indicated *TransportURL*. The SensorTransportGeneric service shall not initiate further connections to the indicated *TransportURL*. The effect on active (in progress) connections to a transport endpoint is implementation dependent, i.e. the TransportGeneric service may either complete or abort in progress transfers.

5.5.2.1 Arguments

Table 5 — Arguments for *DisconnectSensor()*

Argument	Direction	relatedStateVariable
SensorID	*IN*	*A_ARG_TYPE_SensorID*
TransportURL	*IN*	*A_ARG_TYPE_TransportURL*
TransportConnectionID	*IN*	*A_ARG_TYPE_TransportConnectionID*

5.5.2.2 Argument Descriptions

SensorID: The value of this string argument shall identify a (target) sensor which is managed by IoT Management and Control device. See the *SensorID* parameter in IoT Management and Control DataModel service [12] for further details about identifying sensors and their properties.

TransportURL: This argument shall provide the URL of one or more transport connection(s) to the Sensor identified by the *SensorID* argument.

TransportConnectionID: If the value of this argument is an empty string (""), then all active transport connections between the indicated *SensorID* and *TransportURL* shall be disconnected. If the value if this argument identifies a specific transport connection (as returned by the *ConnectSensor()* action), then the specified transport connection between the *SensorID* indicated and *TransportURL* shall be disconnected.

5.5.2.3 Service Requirements

If this IoT Management and Control device implements *DeviceProtection*, then a Control Point identity invoking this action shall have either a *smgt:ConnectSensor* or *Admin* role for the target sensor to successfully complete this action. See the *SensorPermissions* parameter as described in IoT Management and Control DataModel service [12] for further details on the sensor permission model.

5.5.2.4 Control Point Requirements When Calling The Action

None.

5.5.2.5 Dependency on Device State

The SensorTransportGeneric service shall determine if a transport connection is currently active for the *TransportURL* argument (and *TransportConnectionID*) if provided. If an active connection is not found, then error code 704 shall be returned.

5.5.2.6 Effect on Device State

The TransportGeneric service shall not initiate further connections to the indicated *TransportURL*. The effect on active (in progress) connections to a transport endpoint is implementation dependent, i.e. the TransportGeneric service may either complete or abort in progress transfers.

5.5.2.7 Errors

Table 6 — Error Codes for *DisconnectSensor()*

ErrorCode	errorDescription	Description

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture clause on Control.
500-599	TBD	See UPnP Device Architecture clause on Control.
600-699	TBD	See UPnP Device Architecture clause on Control.
702	SensorID not found	The SensorID provided does not correspond to a known sensor.
704	Transport connection not found	The indicated transport connection was not found.

5.5.3 *ReadSensor()*

This allowed action shall return available data records from the sensor identified by the *SensorID* argument limited by the value of the *DataRecordCount* argument.

5.5.3.1 Arguments

Table 7 — Arguments for *ReadSensor()*

Argument	Direction	relatedStateVariable
SensorID	*IN*	*A_ARG_TYPE_SensorID*
SensorClientID	*IN*	*A_ARG_TYPE_SensorClientID*
SensorURN	*IN*	*A_ARG_TYPE_SensorURN*
SensorRecordInfo	*IN*	*A_ARG_TYPE_SensorRecordInfo*
SensorDataTypeEnable	*IN*	*A_ARG_TYPE_SensorDataTypeEnable*
DataRecordCount	*IN*	*A_ARG_TYPE_DataRecordCount*
DataRecords	*OUT*	*A_ARG_TYPE_DataRecords*

5.5.3.2 Argument Descriptions

SensorID: The value of this string argument shall identify a (target) sensor which is managed by IoT Management and Control device. See the *SensorID* parameter in IoT Management and Control DataModel service [12] for further details about identifying sensors and their properties.

SensorClientID: The value of this string argument shall be returned as the value of the DataItem named ClientID by the target sensor. See IoT Management and Control DataModel service [12], Annex B "Required Sensor DataItems" for additional information.

SensorURN: The value of this string argument shall be a URN value identifying a set of DataItems supported by the indicated sensor.

SensorRecordInfo: This argument shall provide a string containing an XML document which identifies the contents of the DataRecord(s) to be provided by the target sensor. See *A_ARG_TYPE_SensorRecordInfo* for further details.

SensorDataTypeEnable: If this argument is set to "1" (one) then each DataRecord <`field`> element shall include type, encoding and namespace (if applicable) attributes for the corresponding DataItem.

Note: Inclusion of DataItem type information is primarily for testing and diagnostic purposes to allow the transport client to verify the expected DataItem types were provided.

DataRecordCount: This argument shall indicate the maximum number of DataRecord(s) to be returned by this action invocation.

DataRecords: This argument shall provide a string containing an XML document which contains DataRecord(s) currently available for the target sensor. See *A_ARG_TYPE_DataRecords* for further details.

5.5.3.3 Service Requirements

If this IoT Management and Control device implements *DeviceProtection*, then a Control Point identity invoking this action shall have either *smgt:ReadSensor* or *Admin* role for the target sensor to successfully complete this action. See the *SensorPermissions* parameter as described in IoT Management and Control DataModel service [12] for further details on the sensor permission model.

5.5.3.4 Control Point Requirements When Calling The Action

None.

5.5.3.5 Dependency on Device State

This action shall complete in a timely manner even if the sensor has no DataRecords to provide.

5.5.3.6 Effect on Device State

DataRecords once read via this SOAP action are not retained by the IoT Management and Control device. However, this action has no impact on transport connections for the target sensor each of which maintain an independent queue of DataRecords.

5.5.3.7 Errors

Table 8 — Error Codes for *ReadSensor()*

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture clause on Control.
500-599	TBD	See UPnP Device Architecture clause on Control.
600-699	TBD	See UPnP Device Architecture clause on Control.
701	Invalid XML argument	An argument value (SensorRecordInfo) provided does not comply with the corresponding schema requirements.
702	SensorID not found	The SensorID provided does not correspond to a known sensor.
703	Sensor URN not found	The SensorURN provided does not correspond to a known URN for the indicated SensorID.
705	Sensor DataItem not found	A DataItem referenced by the SensorRecordInfo XML argument cannot be found.

5.5.4 *WriteSensor()*

This allowed action writes a data record to the sensor identified by the *SensorID* argument.

5.5.4.1 Arguments

Table 9 — Arguments for *WriteSensor()*

Argument	Direction	relatedStateVariable
SensorID	*IN*	*A_ARG_TYPE_SensorID*
SensorURN	*IN*	*A_ARG_TYPE_SensorURN*
DataRecords	*IN*	*A_ARG_TYPE_DataRecords*

5.5.4.2 Argument Descriptions

SensorID: The value of this string argument shall identify a (target) sensor which is managed by IoT Management and Control device. See the *SensorID* parameter in IoT Management and Control DataModel service [12] for further details about identifying sensors and their properties.

SensorURN: The value of this string argument shall be a URN value identifying a set of DataItem(s) supported by the indicated sensor.

DataRecords: This argument shall provide a string containing an XML document which contains DataRecord(s) currently available for the target sensor. See *A_ARG_TYPE_DataRecords* for further details.

5.5.4.3 Service Requirements

If this IoT Management and Control device implements *DeviceProtection*, then a Control Point identity invoking this action shall have either a *smgt:WriteSensor* or *Admin* role for the target sensor to successfully complete this action. See the *SensorPermissions* parameter as described in IoT Management and Control DataModel service [12] for further details on the sensor permission model.

Each DataItem to be written shall allow overall write access as indicated by the DataItem Description Document or DataItem definition for the corresponding SensorURN. Any attempts to write strictly read-only DataItem(s) shall cause the action to fail with error code 706. Writing CSV DataItems which contain a mixture of writable and non-writable fields shall only modify CSV components with explicit values provided. The remaining components of the CSV DataItem shall be silently ignored.

5.5.4.4 Control Point Requirements When Calling The Action

None.

5.5.4.5 Dependency on Device State

The TransportGeneric service may buffer and acknowledge pending writes to an actuator device presuming the pending write operations can be completed in a timely manner. However, if the service cannot buffer further data, error code 707 shall be returned. If the service determines communication with the corresponding sensor is not possible error code 706 shall be returned.

5.5.4.6 Effect on Device State

None.

5.5.4.7 Errors

Table 10 — Error Codes for *WriteSensor()*

ErrorCode	errorDescription	Description
400-499	TBD	See UPnP Device Architecture clause on Control.
500-599	TBD	See UPnP Device Architecture clause on Control.
600-699	TBD	See UPnP Device Architecture clause on Control.
701	Invalid XML argument	An argument value (DataRecords) provided does not comply with the corresponding schema requirements.
702	SensorID not found	The SensorID provided does not correspond to a known sensor.
703	Sensor URN not found	The SensorURN provided does not correspond to a known URN for the indicated SensorID.
705	Sensor DataItem not found	A DataItem referenced by the SensorRecordInfo XML argument cannot be found; or a `<datarecord>` element contains a duplicate DataItem.
706	Sensor DataItem is read-only.	One or more DataItems to be written are marked read-only.
707	Sensor unavailable	The target sensor is disconnected and cannot be successfully written.

5.5.5 *GetSensorTransportConnections()*

This reqired action returns information on current transport connections for the sensor identified by the *SensorID* argument.