

SensorTransportGeneric:1 Service

For UPnP Version 1.0

Status: Standardized DCP (SDCP)

Date: July 1, 2013

Document Version: 1.0

Service Template Version: 2.00

This Standardized DCP has been adopted as a Standardized DCP by the Steering Committee of the UPnP™ Forum, pursuant to Section 2.1(c)(ii) of the UPnP™ Forum Membership Agreement. UPnP™ Forum Members have rights and licenses defined by Section 3 of the UPnP™ Forum Membership Agreement to use and reproduce the Standardized DCP in UPnP™ Compliant Devices. All such use is subject to all of the provisions of the UPnP™ Forum Membership Agreement.

THE UPNP™ FORUM TAKES NO POSITION AS TO WHETHER ANY INTELLECTUAL PROPERTY RIGHTS EXIST IN THE STANDARDIZED DCPS. THE STANDARDIZED DCPS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS". THE UPNP™ FORUM MAKES NO WARRANTIES, EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE STANDARDIZED DCPS, INCLUDING BUT NOT LIMITED TO ALL IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE, OF REASONABLE CARE OR WORKMANLIKE EFFORT, OR RESULTS OR OF LACK OF NEGLIGENCE.

© 2013 UPnP Forum. All Rights Reserved.

Authors ^a	Company
Clarke Stevens	Cablelabs
Jangwook Park (Vice-Chair)	LGE
Paul Jeon (Vice-Chair)	LGE
Russell Berkoff (Chair)	Samsung Electronics
Danilo Santos	Signove
Gerhard Mekenkamp	TPVision
^a The UPnP forum in no way guarantees the accuracy or completeness of this author list and in no way implies any rights for or support from those members listed. This list is not the specifications' contributor list that is kept on the UPnP Forum's website.	

CONTENTS

1	Scope.....	4
2	Normative References.....	4
3	Terms, Definitions and Abbreviations.....	5
4	Notations and conventions.....	5
4.1	Notation	5
4.2	Data Types.....	6
4.3	Vendor-defined Extensions	6
5	Service Modelling Definitions	6
5.1	Service Type	6
5.2	SensorTransportGeneric Service Architecture	6
5.3	Key Concepts	6
5.3.1	Sensor Connectivity.....	6
5.3.2	Sensor HTTP/HTTPS Transport Protocol	7
5.4	State Variables.	8
5.4.1	State Variable Overview.....	8
5.4.2	<u>A ARG TYPE SensorID</u>	8
5.4.3	<u>A ARG TYPE SensorURN</u>	8
5.4.4	<u>A ARG TYPE SensorClientID</u>	8
5.4.5	<u>A ARG TYPE DataRecords</u>	9
5.4.6	<u>A ARG TYPE DataRecordCount</u>	10
5.4.7	<u>A ARG TYPE TransportURL</u>	10
5.4.8	<u>A ARG TYPE SensorDataTypeEnable</u>	10
5.4.9	<u>A ARG TYPE SensorRecordInfo</u>	10
5.4.10	<u>A ARG TYPE TransportConnectionID</u>	11
5.4.11	<u>A ARG TYPE TransportConnections</u>	11
5.5	Actions.....	12
5.5.1	<u>ConnectSensor()</u>	12
5.5.2	<u>DisconnectSensor()</u>	15
5.5.3	<u>ReadSensor()</u>	16
5.5.4	<u>WriteSensor()</u>	17
5.5.5	<u>GetSensorTransportConnections()</u>	19
5.5.6	Error Code Summary	19
6	XML Service Description	20
	Table 1 — State Variables.....	8
	Table 2 — Actions	12
	Table 3 — Arguments for <u>ConnectSensor()</u>	12
	Table 4 — Error Codes for <u>ConnectSensor()</u>	14
	Table 5 — Arguments for <u>DisconnectSensor()</u>	15
	Table 6 — Error Codes for <u>DisconnectSensor()</u>	15
	Table 7 — Arguments for <u>ReadSensor()</u>	16
	Table 8 — Error Codes for <u>ReadSensor()</u>	17

SensorTransportGeneric:1

Table 9 — Arguments for <u>WriteSensor()</u>	17
Table 10 — Error Codes for <u>WriteSensor()</u>	18
Table 11 — Arguments for <u>GetSensorTransportConnections()</u>	19
Table 12 — Error Codes for <u>GetSensorTransportConnections()</u>	19
Table 13 — Error Code Summary.....	20

SensorTransportGeneric: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, which identifies Level 1 of the UPnP SensorTransportGeneric: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 SensorTransportGeneric 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](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] UPnP Sensor and DataStore Architecture Overview, UPnP Forum, July 1, 2013. Available at: <http://www.upnp.org/specs/smgt/UPnP-smgt-SensorManagementArchitectureOverview-v1-20130701.pdf>. Latest version available at: <http://www.upnp.org/specs/smgt/UPnP-smgt-SensorManagementArchitectureOverview-v1.pdf>.

[9] UPnP SensorManagement:1 Device, UPnP Forum July 1, 2013. Available at: <http://www.upnp.org/specs/smgt/UPnP-smgt-SensorManagement-v1-Device-20130701.pdf>. Latest version available at: <http://www.upnp.org/specs/smgt/UPnP-smgt-SensorManagement-v1-Device.pdf>.

[10] UPnP SensorTransportGeneric:1 Service, UPnP Forum July 1, 2013. Available at: <http://www.upnp.org/specs/smgt/UPnP-smgt-SensorTransportGeneric-v1-Service->

[SensorTransportGeneric:1](#)

20130701.pdf. Latest version available at: <http://www.upnp.org/specs/smgt/UPnP-smgt-SensorTransportGeneric-v1-Service.pdf>.

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

[12] UPnP SensorManagement Sensor DataModel Service, UPnP Forum, July 1, 2013. Available at: <http://www.upnp.org/specs/smgt/UPnP-smgt-SensorDataModel-v1-Service-20130701.pdf>. Latest version available at: <http://www.upnp.org/specs/smgt/UPnP-smgt-SensorDataModel-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/drecrecs-v1-20130701.xsd>. Latest version available at: <http://www.upnp.org/schemas/ds/drecrecs.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.

SensorTransportGeneric:1

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 service is used herein to refer to this service type.

5.2 SensorTransportGeneric Service Architecture

The SensorTransportGeneric 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 SensorManagement Sensor DataModel related to the description of sensors supported by this service. See UPnP SensorManagement Sensor DataModel service [12] and UPnP Sensor and DataStore Architecture Overview [8] for additional details.

5.3 Key Concepts

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

5.3.1 Sensor Connectivity

The SensorTransportGeneric 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 SensorManagement Sensor DataModel service [12]. The Sensor 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 SensorTransportGeneric 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 SensorTransportGeneric service

SensorTransportGeneric:1

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

2) SOAP Action Model

The SensorTransportGeneric 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 SensorTransportGeneric 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 SensorManagement 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 SensorTransportGeneric 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:

SensorTransportGeneric:1

<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 |
|---|------------------------|------------------------------|------------|
| <u>A_ARG_TYPE_SensorID</u> | <u>R</u> | <u>string</u> | See 5.4.2 |
| <u>A_ARG_TYPE_SensorURN</u> | <u>R</u> | <u>string</u> | See 5.4.3 |
| <u>A_ARG_TYPE_SensorClientID</u> | <u>R</u> | <u>string</u> | See 5.4.4 |
| <u>A_ARG_TYPE_DataRecords</u> | <u>CR</u> ^b | <u>string</u> (XML fragment) | See 5.4.5 |
| <u>A_ARG_TYPE_DataRecordCount</u> | <u>R</u> | <u>ui4</u> | See 5.4.6 |
| <u>A_ARG_TYPE_TransportURL</u> | <u>R</u> | <u>string</u> | See 5.4.7 |
| <u>A_ARG_TYPE_SensorDataTypeEnable</u> | <u>R</u> | <u>boolean</u> | See 5.4.8 |
| <u>A_ARG_TYPE_SensorRecordInfo</u> | <u>R</u> | <u>string</u> | See 5.4.9 |
| <u>A_ARG_TYPE_TransportConnectionID</u> | <u>R</u> | <u>string</u> | See 5.4.10 |
| <u>A_ARG_TYPE_TransportConnections</u> | <u>R</u> | <u>string</u> | See 5.4.11 |
| ^a <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>O-D</u>). | | | |
| ^b This conditionally required state variable MUST be implemented if actions <u>ReadSensor()</u> or <u>WriteSensor()</u> 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 SensorManagement 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 Dataltems the target sensor can provide. See UPnP SensorManagement Sensor 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 Dataltem named "ClientID".

SensorTransportGeneric:1

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 *SensorTransportGeneric* 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 SensorManagement 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](#)".

SensorTransportGeneric:1

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 SensorManagement 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 SensorDataEnable

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.

SensorTransportGeneric:1

<SensorRecordInfo>

Required. Shall include a namespace declaration for the XML Schema for Sensor DataRecord Information ("urn:schemas-upnp-org:smgt:srcinfo"). 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 Dataltem 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 Dataltem to be included in the <datarecord> generated by the target sensor.

prefix

Allowed. The value provided shall be prefixed to the Dataltem 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

SensorTransportGeneric:1

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.

transportURL

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 |
|--|----------------------------|-----------------------------------|
| <u>ConnectSensor()</u> | R | R |
| <u>DisconnectSensor()</u> | R | R |
| <u>ReadSensor()</u> | A | A |
| <u>WriteSensor()</u> | A | A |
| <u>GetSensorTransportConnections</u> | R | A |
| ^a For a device this column indicates whether the action MUST 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>O-D</u>). | | |
| ^b For a control point this column indicates whether a control point MUST 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>O-D</u>). | | |

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 |
|-----------------------------|-----------|--|
| <u>SensorID</u> | <u>IN</u> | <u>A_ARG_TYPE_SensorID</u> |
| <u>SensorClientID</u> | <u>IN</u> | <u>A_ARG_TYPE_SensorClientID</u> |
| <u>SensorURN</u> | <u>IN</u> | <u>A_ARG_TYPE_SensorURN</u> |
| <u>SensorRecordInfo</u> | <u>IN</u> | <u>A_ARG_TYPE_SensorRecordInfo</u> |
| <u>SensorDataTypeEnable</u> | <u>IN</u> | <u>A_ARG_TYPE_SensorDataTypeEnable</u> |

SensorTransportGeneric:1

| Argument | Direction | relatedStateVariable |
|------------------------------|------------|---|
| <u>TransportURL</u> | <u>IN</u> | <u>A_ARG_TYPE_TransportURL</u> |
| <u>TransportConnectionID</u> | <u>OUT</u> | <u>A_ARG_TYPE_TransportConnectionID</u> |

5.5.1.2 Argument Descriptions

SensorID: The value of this string argument shall identify a (target) sensor which is managed by SensorManagement device. See the SensorID parameter in SensorManagement Sensor 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 SensorManagement Sensor 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 SensorManagement Sensor 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.

SensorTransportGeneric:1

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 SensorManagement 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 SensorManagement Sensor 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 SensorManagement 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 SensorManagement device shall retain the provided TransportURL. The SensorManagement device may connect to the TransportURL to send DataRecord(s) to the sensor transport server endpoint at any time. The SensorManagement 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. |

SensorTransportGeneric:1

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 SensorTransportGeneric service may either complete or abort in progress transfers.

5.5.2.1 Arguments

Table 5 — Arguments for DisconnectSensor()

| Argument | Direction | relatedStateVariable |
|------------------------------|-----------|---|
| <u>SensorID</u> | <u>IN</u> | <u>A_ARG_TYPE_SensorID</u> |
| <u>TransportURL</u> | <u>IN</u> | <u>A_ARG_TYPE_TransportURL</u> |
| <u>TransportConnectionID</u> | <u>IN</u> | <u>A_ARG_TYPE_TransportConnectionID</u> |

5.5.2.2 Argument Descriptions

SensorID: The value of this string argument shall identify a (target) sensor which is managed by SensorManagement device. See the SensorID parameter in SensorManagement Sensor 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 of 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 SensorManagement 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 SensorManagement Sensor 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 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 SensorTransportGeneric service may either complete or abort in progress transfers.

5.5.2.7 Errors

Table 6 — Error Codes for DisconnectSensor()

| ErrorCode | errorDescription | Description |
|-----------|------------------|-------------|
|-----------|------------------|-------------|

SensorTransportGeneric:1

| 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 |
|-----------------------------|------------|--|
| <u>SensorID</u> | <u>IN</u> | <u>A_ARG_TYPE_SensorID</u> |
| <u>SensorClientID</u> | <u>IN</u> | <u>A_ARG_TYPE_SensorClientID</u> |
| <u>SensorURN</u> | <u>IN</u> | <u>A_ARG_TYPE_SensorURN</u> |
| <u>SensorRecordInfo</u> | <u>IN</u> | <u>A_ARG_TYPE_SensorRecordInfo</u> |
| <u>SensorDataTypeEnable</u> | <u>IN</u> | <u>A_ARG_TYPE_SensorDataTypeEnable</u> |
| <u>DataRecordCount</u> | <u>IN</u> | <u>A_ARG_TYPE_DataRecordCount</u> |
| <u>DataRecords</u> | <u>OUT</u> | <u>A_ARG_TYPE_DataRecords</u> |

5.5.3.2 Argument Descriptions

SensorID: The value of this string argument shall identify a (target) sensor which is managed by SensorManagement device. See the SensorID parameter in SensorManagement Sensor 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 Dataltem named ClientID by the target sensor. See SensorManagement Sensor DataModel service [12], Annex B "Required Sensor Dataltems" for additional information.

SensorURN: The value of this string argument shall be a URN value identifying a set of Dataltems 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 Dataltem.

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

SensorTransportGeneric:1

5.5.3.3 Service Requirements

If this SensorManagement 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 SensorManagement Sensor 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 SensorManagement 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 Dataltem not found | A Dataltem 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 |
|--------------------|-----------|-------------------------------|
| <u>SensorID</u> | <u>IN</u> | <u>A_ARG_TYPE_SensorID</u> |
| <u>SensorURN</u> | <u>IN</u> | <u>A_ARG_TYPE_SensorURN</u> |
| <u>DataRecords</u> | <u>IN</u> | <u>A_ARG_TYPE_DataRecords</u> |

5.5.4.2 Argument Descriptions

SensorID: The value of this string argument shall identify a (target) sensor which is managed by SensorManagement device. See the SensorID parameter in SensorManagement Sensor 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 Dataltem(s) supported by the indicated sensor.

SensorTransportGeneric:1

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 SensorManagement 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 SensorManagement Sensor DataModel service [12] for further details on the sensor permission model.

Each Dataltem to be written shall allow overall write access as indicated by the Dataltem Description Document or Dataltem definition for the corresponding SensorURN. Any attempts to write strictly read-only Dataltem(s) shall cause the action to fail with error code 706. Writing CSV Dataltems 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 Dataltem 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 SensorTransportGeneric 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 Dataltem not found | A Dataltem referenced by the SensorRecordInfo XML argument cannot be found; or a <datarecord> element contains a duplicate Dataltem. |
| 706 | Sensor Dataltem is read-only. | One or more Dataltems to be written are marked read-only. |
| 707 | Sensor unavailable | The target sensor is disconnected and cannot be successfully written. |
| | | |

SensorTransportGeneric:1

5.5.5 GetSensorTransportConnections()

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

5.5.5.1 Arguments

Table 11 — Arguments for GetSensorTransportConnections()

| Argument | Direction | relatedStateVariable |
|-----------------------------|------------|--|
| <u>SensorID</u> | <u>IN</u> | <u>A_ARG_TYPE_SensorID</u> |
| <u>TransportConnections</u> | <u>OUT</u> | <u>A_ARG_TYPE_TransportConnections</u> |

5.5.5.2 Argument Descriptions

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

TransportConnections: This argument shall provide a string containing an XML document which contains information about the transport connections currently supported by the indicated sensor.

5.5.5.3 Service Requirements

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

5.5.5.4 Control Point Requirements When Calling The Action

None.

5.5.5.5 Dependency on Device State

This action shall only return information on currently active Sensor transport connections. Prior transport connections which have been cancelled via the DisconnectSensor() action or by the SensorManagement device due to lack of response from the transport endpoint shall not be reported.

5.5.5.6 Effect on Device State

None.

5.5.5.7 Errors

Table 12 — Error Codes for GetSensorTransportConnections()

| 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. |
| | | |

5.5.6 Error Code Summary

The following table lists error codes common to actions for this service type. If an action results in multiple errors, the most specific error should be returned.

Table 13 — Error Code Summary

| 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. |
| 700 | | Reserved for future extensions. |
| 701 | Invalid XML argument | An argument value 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. |
| 704 | Sensor transport connection not found | The indicated transport connection was not found. |
| 705 | Sensor Dataltem not found | A Dataltem referenced by an action cannot be found. |
| 706 | Sensor Dataltem is read-only. | One or more Sensor Dataltems to be written are marked read-only. |
| 707 | Sensor unavailable | The target sensor is disconnected and cannot be successfully written. |
| 708 | Transport connection limit exceeded. | The number of available transport connections to the indicated SensorID has been exceeded. |
| | | |

Note: 800-899 Error Codes are not permitted for standard actions. See UPnP Device Architecture clause on Control for more details.

6 XML Service Description

```
<?xml version="1.0"?>
<scpd xmlns="urn:schemas-upnp-org:service-1-0">

  <specVersion>
    <major>1</major>
    <minor>0</minor>
  </specVersion>

  <actionList>
    <action>
      <name>ConnectSensor</name>
      <argumentList>
        <argument>
          <name>SensorID</name>
          <direction>in</direction>
          <relatedStateVariable>
            A_ARG_TYPE_SensorID
          </relatedStateVariable>
        </argument>
        <argument>
          <name>SensorClientID</name>
          <direction>in</direction>
          <relatedStateVariable>
            A_ARG_TYPE_SensorClientID
          </relatedStateVariable>
        </argument>
        <argument>
          <name>SensorURN</name>
          <direction>in</direction>
```

SensorTransportGeneric:1

```
<relatedStateVariable>
  A_ARG_TYPE_SensorURN
</relatedStateVariable>
</argument>
<argument>
  <name>SensorRecordInfo</name>
  <direction>in</direction>
  <relatedStateVariable>
    A_ARG_TYPE_SensorRecordInfo
  </relatedStateVariable>
</argument>
<argument>
  <name>SensorDataTypeEnable</name>
  <direction>in</direction>
  <relatedStateVariable>
    A_ARG_TYPE_SensorDataTypeEnable
  </relatedStateVariable>
</argument>
<argument>
  <name>TransportURL</name>
  <direction>in</direction>
  <relatedStateVariable>
    A_ARG_TYPE_TransportURL
  </relatedStateVariable>
</argument>
<argument>
  <name>TransportConnectionID</name>
  <direction>out</direction>
  <relatedStateVariable>
    A_ARG_TYPE_TransportConnectionID
  </relatedStateVariable>
</argument>
</argumentList>
</action>

<action>
  <name>DisconnectSensor</name>
  <argumentList>
    <argument>
      <name>SensorID</name>
      <direction>in</direction>
      <relatedStateVariable>
        A_ARG_TYPE_SensorID
      </relatedStateVariable>
    </argument>
    <argument>
      <name>TransportURL</name>
      <direction>in</direction>
      <relatedStateVariable>
        A_ARG_TYPE_TransportURL
      </relatedStateVariable>
    </argument>
    <argument>
      <name>TransportConnectionID</name>
      <direction>in</direction>
      <relatedStateVariable>
        A_ARG_TYPE_TransportConnectionID
      </relatedStateVariable>
    </argument>
  </argumentList>
</action>

<action>
  <name>ReadSensor</name>
  <argumentList>
    <argument>
      <name>SensorID</name>
      <direction>in</direction>
```

SensorTransportGeneric:1

```
<relatedStateVariable>
  A_ARG_TYPE_SensorID
</relatedStateVariable>
</argument>
<argument>
  <name>SensorClientID</name>
  <direction>in</direction>
  <relatedStateVariable>
    A_ARG_TYPE_SensorClientID
  </relatedStateVariable>
</argument>
<argument>
  <name>SensorURN</name>
  <direction>in</direction>
  <relatedStateVariable>
    A_ARG_TYPE_SensorURN
  </relatedStateVariable>
</argument>
<argument>
  <name>SensorRecordInfo</name>
  <direction>in</direction>
  <relatedStateVariable>
    A_ARG_TYPE_SensorRecordInfo
  </relatedStateVariable>
</argument>
<argument>
  <name>SensorDataTypeEnable</name>
  <direction>in</direction>
  <relatedStateVariable>
    A_ARG_TYPE_SensorDataTypeEnable
  </relatedStateVariable>
</argument>
<argument>
  <name>DataRecordCount</name>
  <direction>in</direction>
  <relatedStateVariable>
    A_ARG_TYPE_DataRecordCount
  </relatedStateVariable>
</argument>
<argument>
  <name>DataRecords</name>
  <direction>out</direction>
  <relatedStateVariable>
    A_ARG_TYPE_DataRecords
  </relatedStateVariable>
</argument>
</argumentList>
</action>

<action>
  <name>WriteSensor</name>
  <argumentList>
    <argument>
      <name>SensorID</name>
      <direction>in</direction>
      <relatedStateVariable>
        A_ARG_TYPE_SensorID
      </relatedStateVariable>
    </argument>
    <argument>
      <name>SensorURN</name>
      <direction>in</direction>
      <relatedStateVariable>
        A_ARG_TYPE_SensorURN
      </relatedStateVariable>
    </argument>
    <argument>
      <name>DataRecords</name>
```

SensorTransportGeneric:1

```
<direction>in</direction>
<relatedStateVariable>
  A_ARG_TYPE_DataRecords
</relatedStateVariable>
</argument>
</argumentList>
</action>

<action>
  <name>GetSensorTransportConnections</name>
  <argumentList>
    <argument>
      <name>SensorID</name>
      <direction>in</direction>
      <relatedStateVariable>
        A_ARG_TYPE_SensorID
      </relatedStateVariable>
    </argument>
    <argument>
      <name>TransportConnections</name>
      <direction>out</direction>
      <relatedStateVariable>
        A_ARG_TYPE_TransportConnections
      </relatedStateVariable>
    </argument>
  </argumentList>
</action>
</actionList>

<serviceStateTable>

  <stateVariable sendEvents="no">
    <name>A_ARG_TYPE_SensorID</name>
    <dataType>string</dataType>
  </stateVariable>

  <stateVariable sendEvents="no">
    <name>A_ARG_TYPE_SensorURN</name>
    <dataType>string</dataType>
  </stateVariable>

  <stateVariable sendEvents="no">
    <name>A_ARG_TYPE_SensorClientID</name>
    <dataType>string</dataType>
  </stateVariable>

  <stateVariable sendEvents="no">
    <name>A_ARG_TYPE_DataRecords</name>
    <dataType>string</dataType>
  </stateVariable>

  <stateVariable sendEvents="no">
    <name>A_ARG_TYPE_DataRecordCount</name>
    <dataType>ui4</dataType>
  </stateVariable>

  <stateVariable sendEvents="no">
    <name>A_ARG_TYPE_TransportURL</name>
    <dataType>string</dataType>
  </stateVariable>

  <stateVariable sendEvents="no">
    <name>A_ARG_TYPE_TransportConnectionID</name>
    <dataType>string</dataType>
  </stateVariable>

  <stateVariable sendEvents="no">
```

SensorTransportGeneric:1

```
<name>A_ARG_TYPE_SensorDataTypeEnable</name>
<dataType>boolean</dataType>
</stateVariable>

<stateVariable sendEvents="no">
  <name>A_ARG_TYPE_TransportConnections</name>
  <dataType>string</dataType>
</stateVariable>

</serviceStateTable>
</scpd>
```