DataStore: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.

Authors [a]	Company
Clarke Stevens	Cablelabs
Jangwoong 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

1.1   Introduction

This document defines the service DataStore:1, which identifies Version 1 of the service named DataStore. This Publicly Available Specification is applicable to Standardized DCPs of the UPnP Forum which include this service.

This service definition is compliant with the UPnP Device Architecture, version 1.0.

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: http://www.iso.org (ISO 8601:2004)

[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 SensorManagement Architecture Overview, UPnP Forum, July 1, 2013. Available at: http://www.upnp.org/specs/smgt/UPnP-smgt-SensorManagementArchitecture-v1-20130701.pdf.   Latest   version   available   at: http://www.upnp.org/specs/smgt/UPnP-smgt-SensorManagementArchitecture-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 SensorGenericTransport:1 Service, UPnP Forum July 1, 2013. Available at: http://www.upnp.org/specs/smgt/UPnP-smgt-SensorTransportGeneric-v1-Service-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/ds/UPnP-ds-DataStore-v1-Service-20130701.pdf. Latest version available at: http://www.upnp.org/specs/ds/UPnP-ds-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 DataStore LastChange Eventing, UPnP Forum, July 1, 2013. Available at: http://www.upnp.org/schemas/ds/dsevent-v1-20130701.xsd. Latest version available at: http://www.upnp.org/schemas/ds/dsevent.xsd.

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

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

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

[19] 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.

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

[21] 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

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 XML Schema Part 2: Data Types, Second Edition [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: DataStore:1

DataStore service is used herein to refer to this service type.

5.2 DataStore Service Architecture

The DataStore service provides the ability to acquire and persistently store information for later access. This service allows UPnP devices such as mobile phones and sensors to make information available for subsequent retrieval. This increases the flexibility of the UPnP ecosystem by eliminating requirements to have an immediate nexus between information sources and sinks on the UPnP network. The DataStore service additionally allows UPnP devices with limited or temporary storage capabilities to persist information for subsequent retrieval. The DataStore service constructs are intended to be modelled after and compatible with well-established database models.

The service defined herein provides the following functionality:

- Methods to define, create and delete tables of data records.
- Methods to define and identify the contents a data records.
- Methods to accept data records from both streaming and programmed sources.
- Methods to select and retrieve data record contents.

5.3 Key Concepts

The DataStore service supports a DataStore which is organized as a set of DataTables. Each DataTable consists of a series of DataRecords and a DataTable Dictionary. This section discusses various DataStore constructs and topics to assist the reader in understanding the DataStore service's actions and state variables.

5.3.1   DataTables

A DataTable consists of identifying metadata, a set of DataRecords and a Dictionary. The DataStore service is allowed to provide access control to DataTables via the UPnP DeviceProtection service .

5.3.2   DataTable Dictionary

Each DataTable has an associated Dictionary. The Dictionary is organized as key-value data structure. The values stored in the Dictionary are always of type string, although the string values may contain encoded data types such as XML, Base64. The Dictionary does not provide encoding or type information for key-value pairs stored. The semantics of Dictionary keys are included with the definition of the corresponding DataTable URN. See DataTable URN.

DataTable record fields can refer to key(s) in the DataTable's Dictionary. When a DataTable record is read the DataStore service can automatically resolve Dictionary references by substituting the corresponding the Dictionary value for the Dictionary key contained in the stored DataRecord.

5.3.3   DataTables GUID

Each DataTable is uniquely identified by a GUID. The format of a DataTable's records is defined by an XML document (see XML Schema UPnP DataStore DataTableInfo [17]) which describes DataTable's record contents at the time of DataTable creation. Additional ecosystem specific requirements associated with a DataTable are identified by a URN value associated with the DataTable

5.3.4   DataTable URN

Each DataTable has an associated URN. This URN value assists DataStore clients in determining DataTable contents and to identify sets of related DataTables. The UPnP DataStore service [11] defines a uniform template for generating these normative identifiers.

Two types of DataTable URNs are defined by this specification:

- Sensor Specific

    The DataTable contains recorded output of a Sensor. The URN for this DataTable is identical to the URN (Identifier-Type: smgt-surn) of the Sensor providing DataItems recorded in this DataTable. See the SensorTransportGeneric service [10], section 5.2.6 "Sensor Normative Type Identifiers" for further details.

- Application Specific

    The DataTable contains data generated by home-network application(s) as well as sensor data. The contents of DataRecords for this type of DataTable is determined by the underlying application and the DataTable   URN serves to identify the generating application. The following format shall be used for application specific DataTable URNs.

    ```
    DataTableURN::= urn ":" upnp-org ":" ds-aurn ":" [generic-application-type] ":"
    [application-vendor-identifier] ":" [application-identifier] ":"
    [application-user-identifier] ":" [application-table-identifier]
    ```

The [generic-application-type] field is descriptive and provides an informative classification for the application. For example:

- `Health_and_Wellness`

- `Home_Energy_Management`

- `Home_Security`

The [application-vendor-identifier] shall contain the ICANN domain-name of the organization or vendor which provides the application.

The [application-identifier] shall identify the specific application or family of applications which define the expected DataTable formats.

The [application-user-identifier] field may contain information relating the DataTable to a specific user of an application. This field may be left empty if no such identification is necessary.

The [application-table-identifier] may contain information identifying a specific type of DataTable within a set of related DataTables. This field may be left empty if no such identification is necessary.

5.3.5   DataTable DataRecords

When a DataTable is created (or pre-defined) a DataRecord format is established for the table. Information about each DataTable is conveyed by an XML document including identifying information, DeviceProtection roles, and DataRecord contents (see XML schema UPnP DataStore DataTableInfo [17]). DataRecord(s) consist of fields each defining a DataItem with an associated name and type (see XML schema UPnP DataStore DataRecord [19]).

5.3.6   DataTable DataItems

Each DataTable record consist of a set of DataItem(s). DataItems may include simple scalars, CSVs, abstract data types or XML documents. Each DataItem is characterized by four components: name, type, encoding and an allowed DataItem Description document (see SensorManagement Architecture Overview [8], subclause 4.3, "DataItem Semantics" for a detailed description of DataItems).

5.3.7   DataItem Description Documents

A DataItem may be described by an accompanying XML document (see SensorManagement Architecture Overview [8], subclause 4.4, "DataItem Description XML Document"). A DataStore client should store DataItem Description Documents (if available) in the DataTable Dictionary. When storing these documents in a DataTable Dictionary, the following template shall be used for generating DataTable Dictionary key names:

```
DictionaryKey ::= ItemDescription "-" [DataItem Prefix] DataItem Name
```

For example:

Dictionary KeyName	ItemDescription-[Garage]FreezerTempSetting
Dictionary KeyValue	```<?xml version="1.0" encoding="UTF-8"?>
<DataItemDescription
  xmlns="urn:schemas-upnp-org:sensors:dataitemdescription"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="
    urn:schemas-upnp-org:sensors:did.xsd
    http://www.upnp.org/schemas/sensors/did.xsd"

  itemName="FreezerTempSetting"
  access="rw">

  <description>
    Freezer Temperature Control
  </description>

  <measurement
    units="degC"
    treatment="average"
    accumulation="interval" />

  <limit units="degC" limittype="low" />
  <limit units="degC" limittype="high" />

</DataItemDescription>``` |

5.3.8   DataItem formats

The DataStore service always conveys stored DataItem(s) as strings. However, the contents of the stored strings may be XML documents, integer(s) or binary content conveyed as Base64 strings and the DataRecord format indicates the allowable data types that may be conveyed by these strings. It should be noted that the DataStore service may convert representations. For example an field with type "*uda:r4*" may be correctly conveyed a either 1.001E2 and 100.1. In addition, a DataRecord type may be abstract such as a UPnP **A ARG TYPE** state variable or an "mds:IEEE_11073-ATTR" type representing IEEE-11073 Object Attributes.

5.3.9   DataTables Operations

DataTables are created and deleted via SOAP actions. In addition to DataTable creation, SOAP clients may either directly write DataTable records via SOAP actions or may arrange transport connections via the DataStore service [11] *GetDataStoreTransportURL()* action allowing asynchronous writing of DataTable records. DataTable records are retrieved via SOAP actions. DataTable Dictionary keys are read and written via SOAP actions.

5.3.10   DataTable HTTP/HTTPS Transport Protocol

DataTable connections to URLs provided by the *GetDataStoreTransportURL()* action shall support the HTTP/HTTPS Transport model as follows:

1) When the DataTable transport endpoint has data available, the transport client shall issue an HTTP/HTTPS POST request to the DataStore transport endpoint provided by the TransportURL provided *GetDataStoreTransportURL()*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 DataStore 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 XML Schema DataStore DataRecord Status [21].

Note: This schema is shared with the SensorTransportGeneric service [10].

```xml
<?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 DataStore service DataRecordStatus Schema (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.

5.3.11 DataStore Groups

DataTable(s) within a DataStore shall be a member of (reference) zero or more DataStore groups. DataTable membership in a group indicates the subject DataTable(s) are related. The DataStore service can apply access controls to DataStore groups which are applied to any DataTable(s) referencing these groups. The DataStore service [11], however, only defines membership in a group; additional ecosystem specific requirements may further define additional rules and meaning attached to group membership.

5.3.12 DataStore Events

The DataStore service [11] can notify its clients of state changes via the evented *LastChange* state variable. The *LastChange* state variable includes a summary of events including the creation and deletion of DataTable(s), modifications to DataTable(s) and corresponding Dictionary(s) in addition to the creation of DataStore groups.

5.3.13 DataTable Permissions

The DataStore service [11] may implement the allowed DeviceProtection service [13]. If this feature is implemented, DataStore groups may be assigned permissions. DataTable(s) may allow or restrict access by using the default DeviceProtection roles: *Public* and/or *Basic* or by DataStore group permissions which include: *Master* (Create/Update/Delete), *Read*, and *Write*. The DataStore service [11] defines the syntax for DeviceProtection roles by combining a DataStore group permission component with a DataStore group name (see subclause 5.4, "Device Protection" for a detailed discussion of DataStore group permissions).

5.3.14 DataTable Actions

While no grouping of DataStore actions is defined by this specification, it can be helpful to the reader to consider related DataStore actions together:

- DataTableGroup Management
  - *CreateDataStoreGroups()*
  - *GetDataStoreGroups()*
  - *DeleteDataStoreGroups()*
- DataTable Management
  - *CreateDataStoreTable()*
  - *DeleteDataStoreTable()*
  - *ResetDataStoreTable()*
- DataTable Information
  - *GetDataStoreInfo()*
  - *GetDataStoreTableInfo()*
  - *ModifyDataStoreTable()*
- DataTable Read/Write
  - *ReadDataStoreTableRecords()*
  - *WriteDataStoreTableRecords()*

DataStore:1

- DataTable Properties
    - *CreateDataStoreTableKeyValue()*
    - *GetDataStoreTableKeyValue()*
    - *RemoveDataStoreTableKeyValue()*
- DataTable Transport Connection
    - *GetDataStoreTransportURL()*

5.4   Device Protection

The DataStore service [11] is allowed to restrict control point access to DataTable(s) using the DeviceProtection service [13]. When the DeviceProtection feature is implemented the DataStore service [11] shall support the following roles:

- *Admin* – A control point with the *Admin* role can create/read/write/delete any DataStore Table and can create or remove any DataStore group.

- *Public* – A control point with the *Public* role can read or write specific DataStore tables which permit this access.

- *Basic* – A control point with the *Basic* role can read or write specific DataStore tables which permit this access.

In addition the DataStore service [11] which implements the DeviceProtection feature shall support the following group roles:

- *ds:Master#[GroupName]* - A control point with a *ds:Master#[GroupName]* identity for the indicated DataStore group may create or delete the corresponding group and may create or delete DataTable(s) belonging to that DataStore group. If a created or deleted DataTable participates in multiple DataStore groups, then the control point is required to have corresponding *ds:Master* identities for all groups the target DataTable references.

- *ds:Reader#[GroupName]* - A control point with a *ds:Reader#[GroupName]* identity for the indicated DataStore group may read DataTable(s) which are a member of the identified group.

- *ds:Writer#[GroupName]* - A control point with a *ds:Writer#[GroupName]* identity for the indicated DataStore group may write DataTable(s) which are a member of the identified group.

5.5   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.

5.5.1   State Variable Overview

Table 1 — State Variables

Variable Name	R/A a	Data Type	Allowed Value	Default Value	Eng. Units
LastChange	*R*	**string**	See 5.5.2		
A_ARG_TYPE_DataRecordCount	*R*	**ui4**	See 5.5.3		
A_ARG_TYPE_DataRecordIndex	*R*	**string**	See 5.5.4		
A_ARG_TYPE_DataRecordFilter	*R*	**string**	See 5.5.5		
A_ARG_TYPE_DataTableID	*R*	**string**	See 5.5.6		
A_ARG_TYPE_DataTableInfoFragment	*CR*	**string**	See 5.5.7		
A_ARG_TYPE_DataTableKeyName	*R*	**string**	See 5.5.8		
A_ARG_TYPE_DataTableKeyValue	*R*	**string**	See 5.5.9		
A_ARG_TYPE_DataStoreInfo	*R*	**string**	See 5.5.11		
A_ARG_TYPE_DataTableInfo	*R*	**string**	See 5.5.12		
A_ARG_TYPE_DataTableResetReq	*R*	**boolean**	*See 5.5.13*		
A_ARG_TYPE_DataStoreGroups	*R*	**string**	See 5.5.10		
A_ARG_TYPE_DataRecordPropResolve	*R*	**boolean**	See 5.5.14		
A_ARG_TYPE_DataRecords	*R*	**string**	See 5.5.15		
A_ARG_TYPE_DataRecordsStatus	*R*	**string**	See 5.5.15		
A_ARG_TYPE_DataTransportURL	*R*	**string**	See 5.5.17		
Non-standard state variables implemented by a UPnP vendor go here	*X*	*TBD*	*TBD*	*TBD*	*TBD*
NOTES:					

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

b   CSV stands for Comma-Separated Value list. The type between brackets denotes the UPnP data type used for the elements inside the list. The CSV list concept is defined more formally in the ContentDirectory service template.

c   See referenced subclause for conditions under which the implementation of this state variable is required.

5.5.2   *LastChange*

This required state variable shall contain an XML document which conforms to the XML Schema DataStore LastChange Eventing [15]. This state variable is intended to report events on the DataStore service to clients which have subscribed to the service.

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

  <create>
      <datastoretable
          tableGUID="Created DataTable ID"
          tableURN="Created DataTable URN"
          updateID="Created DataTable UpdateID (0)" />
      <datastoregroup
```

```
                   groupName="Created DataStore Group Name" />

              ...Additional DataStoreTable or DataStoreGroup creates ...

        </create>

        <update>
            <datastoretable
                tableGUID="Updated DataTable ID"
                tableURN="Updated DataTable URN"
                updateType="R|P|G|X|O - Update Type"
                updateID="DataTable Update ID"/>

             ... Additional DataStoreTable updates ...

        </update>

        <delete>
            <datastoretable
                tableGUID="Deleted DataTable ID"
                tableURN="Deleted DataTable URN"
                updateID="Deleted DataTable UpdateID (last)"/>
            <datastoregroup
                groupName="Deleted DataStore Group Name" />

             ... Additional DataStoreTable or DataStoreGroup deletes ...

        </delete>

</StateEvent>
```

`<?xml>`

Allowed. Case sensitive.

`<StateEvent>`

Required. Shall include a namespace declaration for the XML Schema DataStore LastChange Eventing ("urn:schemas-upnp-org:ds:dsevent"). Shall include zero or more of the following elements. This namespace defines the following elements and attributes:

`<create>`

Allowed. Indicates creation of a DataTable or DataStore group. Includes one or more of the following elements and attributes for DataTable(s) managed by this DataStore service:

`<datastoretable>`

Conditionally Required. Identifies the DataStore table created. This element shall be included when a new DataTable is created.

`tableGUID`

Required. Identifies GUID of the DataStore table created.

`tableURN`

Required. Identifies the URN of the DataStore table created.

`updateID`

Required. Identifies the initial updateID for the DataStore table.

`<datastoregroup>`

Conditionally Required. Identifies the DataStore group created. This element shall be included when a new DataStoreGroup is created.

`groupName`

Required. Identifies DataStore group name being created.

`<update>`

Allowed. Indicates update to a DataTable. Includes one or more of the following elements and attributes for DataTable(s) managed by this DataStore service:

<datastoretable>

Conditionally Required. Identifies the DataStore table updated. This element shall be included when DataTable updates occur.

tableGUID

Required. Identifies GUID of the DataStore table updated.

tableURN

Required. Identifies the URN of the DataStore table updated.

updateType

Required. Identifies the type(s) of DataTable updates. CSV of allowed values: "*R*" – Records, "*P*" – Properties, "*G*" – Groups or Permissions, "*X*" - DataTable reset or "*O*" - Other DataTable attributes.

updateID

Required. Identifies the updateID for the DataTable. (See discussion below).

<delete>

Allowed. Indicates removal of a DataTable. Includes one or more of the following elements and attributes for DataTable(s) managed by this DataStore service:

<datastoretable>

Conditionally Required. Identifies the DataStore table deleted. This element shall be included when a DataTable is deleted.

tableGUID

Required. Identifies GUID of the DataStore table deleted.

tableURN

Required. Identifies the URN of the DataStore table deleted.

updateID

Required. Identifies the final updateID for the DataStore table.

<datastoregroup>

Conditionally Required. Identifies the DataStore group deleted. This element shall be included when a DataStoreGroup is deleted.

groupName

Required. Identifies DataStore group name being deleted.

The *LastChange* state variable is subject to event moderation. Updates to the DataStore LastChange Eventing Document are appended to the document in order of occurrence. To reduce the size of the *LastChange* state variable document, consecutive `update` elements to the same DataTable should be combined. In this case the value of the `updateID` element shall reflect the most recent DataTable update. Once the *LastChange* state variable is transmitted to subscribed clients, its contents are reset to contain only the `<StateEvent>` element.

5.5.3   *A_ARG_TYPE_DataRecordCount*

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

5.5.4   *A_ARG_TYPE_DataRecordIndex*

This required state variable shall provide a **string** argument that identifies a position within a DataTable. A value of ("*0*") indicates first available record in the DataTable. The syntax and semantics of other DataTable index values are determined by the implementation.

5.5.5   *A_ARG_TYPE_DataRecordFilter*

This required state variable shall define a **string** argument that contains an XML document which conforms to the XML Schema UPnP DataStore DataRecordFilter [20]. This defines criteria to qualify for DataRecord(s) for inclusion in the associated action result.

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

    <filterset>
        <filter
            condition="Filter condition expression" />

        ... Additional <filter> elements ...

    </filterset>

    ... Additional <filterset> elements ...

</DataRecordFilter>
```

`<?xml>`

> Allowed. Case sensitive.

> > `<DataRecordFilter>`

> > > Required. Shall include a namespace declaration for the XML Schema UPnP DataStore DataRecordFilter ("urn:schemas-upnp-org:ds:dsfilter"). Shall include the following elements.

> > > > `<filterset>`

> > > > > Allowed. This element may be specified zero or more times. If all of the `<filter>` element conditions specified within the scope of any `<filterset>` element are met, then the DataRecord shall be included in the corresponding action result. This element shall Include one or more of the following elements:

> > > > > > `<filter>`

> > > > > > > Required. Defines a condition which shall be met to include a DataRecord the corresponding action result.

> > > > > > > > `condition`

> > > > > > > > > Required. xsd:string. A string defining a test condition. See Table 2 for allowed values.

Table 2 — Allowed filter `condition` attribute values

DataItem	Condition	Value
ReceiveTimeStamp	>, <, =	**xsd:dateTime**
ReceiveTimeStamp	>	**xsd:duration**
ObservationTimeStamp	>, <, =	**xsd:dateTime**
ObservationTimeStamp	>	**xsd:duration**
ClientID	=	**string**
<<Any supported DataItem>>	IS NULL, IS NOT NULL	none

Examples:

Obtain DataRecords received during April 9, 2013:

```xml
<DataRecordFilter>
    <filterset>
        <filter condition="ReceiveTimeStamp > 2013-04-09T00:00:00" />
```

```
      <filter condition="ReceiveTimeStamp < 2013-04-09T23:59:59" />
   <filterset>
</DataRecordFilter>
```

Obtain DataRecords received during from one hour ago:

```
<DataRecordFilter>
   <filterset>
      <filter condition="ReceiveTimeStamp > "P1D" />
   </filterset>
<DataRecordFilter>
```

Obtain DataRecords containing Medical Device (Handle 1) observations recorded within the last hour:

```
<DataRecordFilter>
   <filterset>
      <filter condition="$1-Obs IS NOT NULL" />
      <filter condition="ObservationTimeStamp > "PT1H" />
   </filterset>
</DataRecordFilter>
```

5.5.6   *A_ARG_TYPE_DataTableID*

This required state variable shall define a **string** argument which identifies an instance of a DataTable within a DataStore. This argument shall be identical to the `tableGUID` attribute value in the UPnP DataStore DataStoreInfo XML document returned by the *GetDataStoreInfo()* action for the corresponding DataTable.

5.5.7   *A_ARG_TYPE_DataTableInfoFragment*

This conditionally required state variable shall be implemented if the *ModifyDataTableInfo()* action is implemented. This state variable defines a **string** argument that contains an XML fragment from the XML Schema UPnP DataStore DataTableInfo [17]. See the *ModifyDataTableInfo()* action description for specific XML top level elements which are allowed.

5.5.8   *A_ARG_TYPE_DataTableKeyName*

This required state variable shall define a **string** argument that identifies a DataTable Dictionary key.

5.5.9   *A_ARG_TYPE_DataTableKeyValue*

This required state variable shall define a **string** argument that provides the value of a DataTable Dictionary entry corresponding to a DataTable Dictionary key.

5.5.10   *A_ARG_TYPE_DataStoreGroups*

This conditionally required state variable shall be implemented if the DataStore group actions *CreateDataStoreGroup()*, *DeleteDataStoreGroups()* are implemented. This state variable shall define a **string** argument that contains an XML document which conforms to the XML Schema UPnP DataStore DataStoreGroups [18].

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

   <datastoregroup
      groupName="Name of DataStore group" />

   ... Additional <datastoregroup> elements ...
```

```
    </DataStoreGroups>
```

`<?xml>`

Allowed. Case sensitive.

> `<DataStoreGroups>`
>
> > Required. XML. Shall include a namespace declaration for the XML Schema UPnP DataStore DataStoreGroups ("urn:schemas-upnp-org:ds:dsgroups"). Shall include zero or more of the following element:
> >
> > > `<datastoregroup>`
> > >
> > > > Allowed. XML. Each element shall provide a single DataStore group. See `groupName` attribute.
> > > >
> > > > > `groupName`
> > > > >
> > > > > > Required. xsd:string. The value of this attribute is an existing DataStore group name.

5.5.11   *A_ARG_TYPE_DataStoreInfo*

This required state variable shall define a **string** argument that contains an XML document which conforms to the XML Schema UPnP DataStore DataStoreInfo [16]. This document is used to identify DataStore names and associated DataTable(s) managed by the DataStore service.

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

   <datastoretables>

      <datastoretable
         tableGUID="DataTable ID"
         tableURN="DataTable URN"
         updateID="DataTable Update ID">
      </datastoretable>

      ... Additional DataStoreTable(s) ...

   </datastoretables>
</DataStoreInfo>
```

`<?xml>`

Allowed. Case sensitive.

> `<DataStoreInfo>`
>
> > Required. Shall include a namespace declaration for the XML Schema UPnP DataStore DataStoreInfo ("urn:schemas-upnp-org:ds:dsinfo"). Shall include the following elements.
> >
> > > `<datastoretables>`
> > >
> > > > Allowed. XML. Includes one or more of the following elements listing DataTable(s) managed by this DataStore service:
> > > >
> > > > > `<datastoretable>`
> > > > >
> > > > > > Allowed. XML. This element defines the properties of a DataTable supported by this DataStore service. Contains the following attributes:
> > > > > >
> > > > > > > `tableGUID`
> > > > > > >
> > > > > > > > Required. xsd:string. Each DataTable within a DataStore shall have a unique tableGUID value as indicated by this attribute.

tableURN

> Required. xsd:string. Each DataTable is allowed to specify a URN value as indicated by this attribute. See subclause 5.3.4, "DataTable URN" for permissible URN contents. This URN value shall identify the format of all DataRecord(s) contained in the corresponding DataTable. Separate DataTable(s) are permitted to use the same URN values which indicates that the DataTable(s) share a common DataRecord format.

updateID

> Required. xsd:unsignedInt. Identifies the current update ID for the DataTable.

5.5.12 *A_ARG_TYPE_DataTableInfo*

This required state variable shall define a **string** argument that contains an XML document which conforms to the XML Schema UPnP DataStore DataTableInfo [17]. This document is used to identify and define the properties of a DataTable within a DataStore.

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

    tableGUID="ID of DataTable"
    tableURN="URN of DataTable"
    updateID="UpdateID of DataTable"
>

    <datatablegroups>
        <datastoregroup groupName="Name of DataStore group" />

        ... Additional <datastoregroup> elements ...

    </datatablegroups>

    <datatableroles>
        <datatablerole name="Public | Basic - Dev Protection builtin roles">
            Read,Write - CSV of DataStore table permissions
        </datatablerole>

        ... Additional <datatablerole> elements ...

    </datatableroles>

    <datatableretain   count="Max DataRecord Count"
                    duration="Max DataRecord Duration" />

    <datarecord>
        <field    name="Name of DataItem"
                type="Type of DataItem"
                encoding="Encoding of DataItem"
                required="0|1 1 = DataItem is mandatory"
                namespace="namespace(For XML-based DataItem)"
                tableprop="0|1 1 = DataItem refers to DataTable Dictionary"
        />

        ... Additional <field> elements ...

    </datarecord>

</DataTableInfo>
```

`<?xml>`
   Allowed. Case sensitive.

DataStore:1

`<DataTableInfo>`

Required. XML. Shall include a namespace declaration for the DataStore service DataTableInfo Schema [17] ("urn:schemas-upnp-org:ds:dtinfo"). Shall include the following elements.

`tableGUID`

Required. xsd:string. Each DataTable within a DataStore shall have a unique tableGUID value as indicated by this attribute. When a DataTable is being created, the `tableGUID` attribute shall be empty.

`tableURN`

Required. xsd:string. Each DataTable is allowed to specify a URN value as indicated by this attribute. See subclause 5.3.4, "DataTable URN" for permissible URN contents. This URN value shall identify the format of all DataRecord(s) contained in the corresponding DataTable. Separate DataTable(s) are permitted to use the same URN values which indicates that the DataTable(s) share a common DataRecord format.

`updateID`

Required. xsd:unsignedInteger. Identifies the current updateID for the DataTable. When a DataTable is being created, the `updateID` attribute shall be ("0").

`<datatablegroups>`

Allowed. XML. Contains zero or more elements indicating DataStore groups this DataTable is a member of. Contains the following elements:

`<datastoregroup>`

Allowed. xsd:string. Each element shall provide a single DataStore group the DataTable is a member of. See `groupName` attribute.

`groupName`

Required xsd:string. The value of this attribute is an existing DataStore group which this table is (to be) a member of.

`<datatableroles>`

Allowed. XML. Contains zero or more elements indicating access privileges for non-group specific control point roles. Contains the following elements:

`<datatablerole>`

Allowed. xsd:string. The value of this element is a CSV list of allowable actions for the control point role indicated by the `name` attribute. The allowed values of this element are: "*Read*", "*Write*".

`name`

Required. xsd:string. The value of this attribute a control point device protection role. The allowed values for this attribute are: "*Public*" or "*Basic*".

`<datatableretain>`

Allowed. XML. Defines the properties of a DataTable supported by this DataStore service. Contains the following attributes:

`count`

Required. xsd:integer. This attribute specifies the maximum number of DataRecord(s) to be retained in the DataTable. When this count is exceeded, the oldest DataRecord(s) present in the indicated DataTable shall be discarded. A value of **0** for the `count` attribute indicates that the number of retained DataRecord(s) is not limited. In order to facilitate efficient implementation of this requirement, a DataTable is permitted to exceed the indicated number of DataRecord(s) for a period not to exceed 60s.

`duration`

Required. xsd:duration. specifies the maximum age for a given DataRecord. DataRecord(s) present in the DataTable which have exceeded the indicated duration shall be discarded. In order to facilitate efficient implementation of this requirement, a DataTable is permitted to retain otherwise expired DataRecord(s) for a period not to exceed 600s.

```
<datarecord>
```

> Allowed. XML. This element defines the format for all DataRecord(s) contained in a given DataTable. The format of a DataRecord is specified by following included `<field>` elements which define individual DataItem(s) in a DataRecord.:

```
<field>
```

>> Required. XML. This element defines an individual DataItem(s) in a DataRecord. Contains the following attributes:

>>> name

>>>> Required. xsd:string. This attribute shall provide a unique name for each DataItem within a DataRecord

>>> type

>>>> Required. xsd:string. This attribute shall provide data type information for each DataItem within a DataRecord. See SensorManagement Architecture Overview [8], subclause 4.3.3 "DataItem Type" for further information.

>>> 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*". See SensorManagement Architecture Overview [8], subclause 4.3.4 "DataItem Encoding" for further information.

>>> required

>>>> Allowed. xsd:boolean. This attribute shall indicate whether a field element corresponding to a DataItem must be present in DataRecord submitted to a DataStoreTable. An attribute value of ("*1*") indicates that a DataRecord shall contain a field element for the corresponding DataItem. An attribute value of ("*0*") indicates that a DataRecord is permitted to omit the corresponding DataItem.

>>> 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 root element. 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.

>>> tableprop

>>>> Allowed. xsd:boolean. This attribute shall indicate that the corresponding DataItem value contains to a DataTable Dictionary key. The DataStore service can be requested to automatically resolve this DataTable Dictionary key reference by using the DataItem value to lookup the corresponding DataTable Dictionary entry. The value of this Dictionary entry is then returned as the DataItem value when the corresponding DataRecord is read.

5.5.13 *A_ARG_TYPE_DataTableResetReq*

This required state variable shall define a boolean value which enables various DataTable reset operations for the *ResetDataStoreTable()* action.

5.5.14 *A_ARG_TYPE_DataRecordPropResolve*

This required state variable shall define a boolean value. If this argument is ("*1*") then the DataStore service shall attempt to resolve any DataRecord fields defined with the `tableprop` attribute by using the corresponding DataItem value as a DataTable Dictionary key. The returned DataTable record shall contain the corresponding DataTable Dictionary key value. If the named DataTable Dictionary Key is missing the corresponding returned DataItem shall be "null". If this argument is ("*0*") then the unresolved DataTable Dictionary key is returned for DataRecord fields defined with the `tableprop` attribute.

5.5.15 *A_ARG_TYPE_DataRecords*

This required state variable shall define a string argument that contains an XML document which conforms to the XML Schema UPnP DataStore DataRecord [19]. This document is used to submit and retrieve DataRecord(s) from a DataStore service.

```xml
<?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.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. XML. 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. XML. 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, See SensorManagement Architecture Overview [8], subclause 4.3.1, "DataItem Name" for encoding of the `name` attribute.

>>>>> type

>>>>>> Allowed. xsd:string. This attribute shall provide data type information for each DataItem within a DataRecord. See SensorManagement Architecture Overview [8], subclause 4.3.3, "DataItem Type" 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*". See SensorManagement Architecture Overview [8], subclause 4.3.4, "DataItem Encoding" for encoding of the `encoding` attribute.

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.

5.5.16   *A_ARG_TYPE_DataRecordsStatus*

This required state variable shall define a string argument that contains an XML document which conforms to the XML Schema UPnP DataStore DataRecord Status [21] or an empty string. This document shall be generated when one or more DataRecords were not accepted. If all DataRecords were accepted, then an empty string shall be returned by this argument.

```xml
<?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 input <datarecord> provided -->
   <datarecordstatus accepted="0|1" />

   ... Additional <datarecordstatus> elements ...

</DataRecordsStatus>
```

`<?xml>`

Required. Case Sensitive.

`<DataRecordsStatus>`

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

`<datarecordstatus>`

Required. XML. For each `<datarecord>` element in the original 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.

5.5.17   *A_ARG_TYPE_DataTransportURL*

This required state variable shall define a **string** argument that conforms to Uniform Resource Locator syntax (see HyperText Transport Protocol – HTTP/1.1 [4]). The purpose of this URL is to enable clients of the DataStore service to submit asynchronous updates to a DataTable that is associated with the indicated URL value. See *GetDataStoreTransportURL()* action for further protocol requirements.

5.6 Eventing and Moderation

Table 3 — Eventing and Moderation

Variable Name	Evented	Moderation	
		Moder-ated a	Criteria
LastChange	*YES*	*YES*	Min 0.2 seconds between events
a *YES* = The state variable MUST be moderated with the criteria			

5.6.1 Eventing of *LastChange*

The *LastChange* state variable is evented and moderated according to the GENA eventing mechanism as defined by the UPnP Device Architecture, version 1.0 [1]. When multiple object modifications occur within the same moderation period (as determined by the implementation), each change shall be accumulated in the *LastChange* state variable and shall be evented as a single event notification message after the current moderation period expires. After the event notification message has been sent to all subscribed control points, the value of the *LastChange* state variable is reset when an update to the *LastChange* state variable becomes necessary; that is: when the next event occurs. The resulting value is a fresh *LastChange XML Document* that contains a single element that represents the update (that is: it contains the first update event following the distribution of the previous event message to all subscribers). Subsequently, additional update elements are added to the LastChange XML Document until the current moderation period ends and the current value of the *LastChange* state variable (i.e. the current event message) is propagated to all event subscribers.

The *LastChange* state variable is not required to accumulate changes when the DataStore service is *off-line* nor when the DataStore service has no subscribers for events. When the DataStore service comes *on-line*, the *LastChange* state variable may be empty. It is not required to event changes that had been accumulated but not evented when the DataStore service last went *off-line*.

5.7 Actions

Table 4 — Actions

Name	Device R/A a	Control Point R/A b
CreateDataStoreGroups()	CA	A
CreateDataStoreTable()	CA	A
DeleteDataStoreGroups)	CR	A
DeleteDataStoreTable()	CR	A
GetDataStoreTableKeyValue()	R	A
GetDataStoreGroups()	R	A
GetDataStoreInfo ()	R	R
GetDataStoreTableInfo()	R	R
GetDataStoreTransportURL()	R	A
ModifyDataStoreTable()	A	A
ReadDataStoreTableRecords()	R	A
RemoveDataStoreTableKeyValue()	CR	A
ResetDataStoreTable()	R	A
SetDataStoreTableKeyValue()	CA	A
WriteDataStoreTableRecords()	R	A

Name	Device R/A a	Control Point R/A b
a For a device this column indicates whether the action MUST be implemented or not, where *R* = REQUIRED, *O* = 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*).		

5.7.1   *CreateDataStoreGroups()*

This conditionally allowed action declares new DataStore group name(s). A DataStore implementation may only employ predefined DataStore groups in which case this action shall not be implemented.

5.7.1.1   Arguments

Table 5 — Arguments for *CreateDataStoreGroups()*

Argument	Direction	relatedStateVariable
DataStoreGroupList	*IN*	*A_ARG_TYPE_DataStoreGroups*

5.7.1.2   Argument Descriptions

DataStoreGroupList: Provides an XML document [18] describing listing DataStore group identifiers to be declared.

5.7.1.3   Service Requirements

The DataStore service shall return error 704 if a declared group identifier already exists. Additionally, the DataStore service may reserve group names or group name prefixes based on external ecosystem requirements in which case error 715 shall be returned. This action shall make no changes to the DataStore unless all requested DataStore groups can be created.

5.7.1.4   Control Point Requirements When Calling The Action

A control point identity invoking this action must have sufficient privileges to create DataStore groups.

5.7.1.5   Dependency on Device State

None.

5.7.1.6   Effect on Device State

A *LastChange* event notification shall be generated due to this action. See LastChange state variable.

5.7.1.7   Errors

Table 6 — Error Codes for *CreateDataStoreGroups()*

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	The (DataStoreGroupList) argument XML document is not valid.
703	Insufficient role(s) or permission(s).	The control point does not have sufficient roles to perform the requested action.

ErrorCode	errorDescription	Description
704	Invalid group(s).	The DataStore group(s) already exist or are reserved by the implementation.
715	Reserved group name.	The DataStore group name has been reserved for use by the implementation.

5.7.2 *CreateDataStoreTable()*

This conditionally allowed action creates tables a new DataTable. A DataStore implementation may only employ predefined DataStore tables in which case this action shall not be implemented.

5.7.2.1 Arguments

Table 7 — Arguments for *CreateDataStoreTable()*

Argument	Direction	relatedStateVariable
DataTableInfo	*IN*	*A_ARG_TYPE_DataTableInfo*
DataTableID	*OUT*	*A_ARG_TYPE_DataTableID*

5.7.2.2 Argument Descriptions

DataTableInfo: Provides a UPnP DataStore DataTableInfo XML document [17] describing the DataTable to be created in the indicated DataStore.

DataTableID: Returns a unique GUID assigned to the newly created DataTable. This GUID value shall also be identical to the `tableGUID` attribute value in the UPnP DataStore DataStoreInfo XML document returned by the *GetDataStoreInfo()* action.

5.7.2.3 Service Requirements

If this action is implemented, then the *DeleteDataStoreTable()* action shall also be implemented.

5.7.2.4 Control Point Requirements When Calling The Action

A control point identity invoking this action must have sufficient privileges to create a DataTable in the target DataStore. A control point identity with an *Admin* role may create any DataTable subject to any additional device ecosystem constraints. Otherwise, the control point must have a *ds:Master#[GroupName]* role for any DataStore groups indicated in the `<datatablegroups>` element of the UPnP DataStore DataTableInfo XML document.

5.7.2.5 Dependency on Device State

None.

5.7.2.6 Effect on Device State

A LastChange event notification shall be generated due to this action. See LastChange state variable.

5.7.2.7 Errors

Table 8 — Error Codes for *CreateDataStoreTable()*

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	The (DataTableInfo) argument XML document is not valid.
703	Insufficient role(s) or permission(s).	The control point does not have sufficient roles to perform the requested action.

ErrorCode	errorDescription	Description
704	Invalid group(s).	The DataStore group(s) do not exist.
705	Invalid role(s) or permission(s).	The DataTable Role(s) or Permission(s) do not exist.

5.7.3 *DeleteDataStoreGroups()*

This conditionally required action deletes DataStore groups. This action shall be implemented if the *CreateDataStoreGroups()* action is supported.

5.7.3.1 Arguments

Table 9 — Arguments for *DeleteDataStoreGroups()*

Argument	Direction	relatedStateVariable
DataStoreGroupList	*IN*	*A_ARG_TYPE_DataStoreGroups*

5.7.3.2 Argument Descriptions

DataStoreGroupList: Provides a UPnP DataStore DataStoreGroups XML document [18] listing DataStore groups to be deleted.

5.7.3.3 Service Requirements

A control point identity invoking this action must have sufficient privileges to create DataStore groups. This action shall fail with error code 710 if the DataStore contains any DataTable(s) which reference any of the DataStore groups to be deleted.

5.7.3.4 Control Point Requirements When Calling The Action

A control point identity invoking this action must have sufficient privileges to create a DataTable in the target DataStore. A control point identity with an *Admin* role may delete any DataStore group subject to any additional device ecosystem constraints.

5.7.3.5 Dependency on Device State

None.

5.7.3.6 Effect on Device State

A LastChange event notification shall be generated due to this action. See LastChange state variable.

5.7.3.7 Errors

Table 10 — Error Codes for *DeleteDataStoreGroups()*

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	The (DataStoreGroupList) argument XML document is not valid.
703	Insufficient role(s) or permission(s).	The control point does not have sufficient roles to perform the requested action.
704	Invalid group(s).	The DataStore group(s) do not exist.
710	Groups in use	The DataStore group(s) cannot be modified or deleted since they are currently referenced by DataTable(s).

5.7.4 *DeleteDataStoreTable()*

This conditionally required action deletes tables an existing DataTable. This action shall be implemented if the *CreateDataStoreTable()* action is supported.

5.7.4.1 Arguments

Table 11 — Arguments for *DeleteDataStoreTable()*

Argument	Direction	relatedStateVariable
DataTableID	*IN*	*A_ARG_TYPE_DataTableID*

5.7.4.2 Argument Descriptions

DataTableID: This argument shall provide an identifier associated with the target DataTable.

5.7.4.3 Service Requirements

A control point identity invoking this action must have sufficient privileges to delete a DataTable in the target DataStore. A control point identity with an *Admin* role may delete any DataTable subject to any additional device ecosystem constraints. Otherwise, the control point must have a *ds:Master#[GroupName]* role for any DataStore groups indicated in the `<datatablegroups>` element of the DataTable. The DataStore service shall insure that any existing DataTable connections are correctly terminated. Specifically, transport connections associated with the deleted DataTable shall be closed after returning status for any pending operations. Previously issued transport URLs associated with the DataTable shall be invalidated.

5.7.4.4 Control Point Requirements When Calling The Action

A control point identity invoking this action shall have sufficient privileges to delete the indicated DataTable.

5.7.4.5 Dependency on Device State

None.

5.7.4.6 Effect on Device State

A LastChange event notification shall be generated due to this action. See LastChange state variable.

5.7.4.7 Errors

Table 12 — Error Codes for *DeleteDataStoreTable()*

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	DataTable not found	The DataTable indicated by DataTableID cannot be found.
703	Insufficient role(s) or permission(s).	The control point does not have sufficient roles to perform the requested action.

5.7.5 *GetDataStoreTableKeyValue()*

This required action returns the requested DataTable Dictionary key value for the indicated DataTable.

5.7.5.1   Arguments

Table 13 — Arguments for *GetDataStoreTableKeyValue()*

Argument	Direction	relatedStateVariable
DataTableID	*IN*	*A_ARG_TYPE_DataTableID*
DataTableKeyName	*IN*	*A_ARG_TYPE_DataTableKeyName*
DataTableKeyValue	*OUT*	*A_ARG_TYPE_DataTableKeyValue*

5.7.5.2   Argument Descriptions

DataTableID: This argument shall provide an identifier associated with the target DataTable.

DataTableKeyName: This argument identifies the DataTable Dictionary key value to be returned.

DataTableKeyValue: This argument returns the value of the requested DataTable Dictionary key.

5.7.5.3   Service Requirements

The DataStore service shall verify that the requesting control point identity has sufficient privileges to read the referenced DataTable.

5.7.5.4   Control Point Requirements When Calling The Action

A control point identity invoking this action shall have sufficient privileges to read the indicated DataTable.

5.7.5.5   Dependency on Device State

None.

5.7.5.6   Effect on Device State

None.

5.7.5.7   Errors

Table 14 — Error Codes for *GetDataStoreTableKeyValue()*

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	DataTable not found	The DataTable indicated by DataTableID cannot be found.
703	Insufficient role(s) or permission(s).	The control point does not have sufficient roles to perform the requested action.
707	Key name not found	The indicated DataTable Dictionary KeyName was not found.

5.7.6   *GetDataStoreGroups()*

This required action returns an XML document [18] listing currently defined DataStore group identifiers.

5.7.6.1   Arguments

Table 15 — Arguments for *GetDataStoreGroups()*

Argument	Direction	relatedStateVariable

DataStore:1

Argument	Direction	relatedStateVariable
DataStoreGroupList	*OUT*	*A_ARG_TYPE_DataStoreGroups*

5.7.6.2 Argument Descriptions

DataStoreGroupList: This argument returns a UPnP DataStore DataStoreGroups XML document [18] containing DataStore groups.

5.7.6.3 Service Requirements

If DeviceProtection is implemented, then this action shall only include DataStore group identifiers the invoking control point identity can access. Otherwise, this action shall return all available DataStore group identifiers currently declared.

5.7.6.4 Control Point Requirements When Calling The Action

None.

5.7.6.5 Dependency on Device State

None.

5.7.6.6 Effect on Device State

None.

5.7.6.7 Errors

Table 16 — Error Codes for *GetDataStoreGroups()*

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.

5.7.7 *GetDataStoreInfo()*

This required action returns information about DataStore DataTable(s) currently supported by this DataStore service.

5.7.7.1 Arguments

Table 17 — Arguments for *GetDataStoreInfo()*

Argument	Direction	relatedStateVariable
DataStoreInfo	*OUT*	*A_ARG_TYPE_DataStoreInfo*

5.7.7.2 Argument Descriptions

DataStoreInfo: This argument returns a UPnP DataStore DataStoreInfo XML document [16] describing DataTable(s) supported by this DataStore service.

5.7.7.3 Service Requirements

If DeviceProtection is supported, then the DataStore service shall only return information about DataTable(s) which the invoking control point identity has access to.

5.7.7.4 Control Point Requirements When Calling The Action

None.

5.7.7.5 Dependency on Device State

None.

DataStore:1

5.7.7.6 Effect on Device State

None.

5.7.7.7 Errors

Table 18 — Error Codes for *GetDataStoreInfo()*

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.

5.7.8 *GetDataStoreTableInfo()*

This required action shall return an XML document containing information about the DataTable associated the *DataTableID* argument.

5.7.8.1 Arguments

Table 19 — Arguments for *GetDataStoreTableInfo()*

Argument	Direction	relatedStateVariable
DataTableID	*IN*	*A_ARG_TYPE_DataTableID*
DataTableInfo	*OUT*	*A_ARG_TYPE_DataTableInfo*

5.7.8.2 Argument Descriptions

DataTableID: This argument shall identify a DataTable associated with the value of this argument

DataTableInfo: This argument shall return a UPnP DataStore DataTableInfo XML document [17] describing the DataTable identified by the *DataTableID* argument.

5.7.8.3 Service Requirements

If DeviceProtection is supported, this action shall only succeed if the invoking control point identity has sufficient access privileges.

5.7.8.4 Control Point Requirements When Calling The Action

None.

5.7.8.5 Dependency on Device State

None.

5.7.8.6 Effect on Device State

None.

5.7.8.7 Errors

Table 20 — Error Codes for *GetDataStoreTableInfo()*

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	DataTable not found	The DataTable indicated by DataTableID cannot be found.
703	Insufficient role(s) or permission(s).	The control point does not have sufficient roles to perform the requested action.

DataStore:1

ErrorCode	errorDescription	Description

5.7.9 *GetDataStoreTransportURL()*

This required action shall return a URL which provides streaming write access to the identified DataTable.

5.7.9.1 Arguments

Table 21 — Arguments for *GetDataStoreTransportURL()*

Argument	Direction	relatedStateVariable
DataTableID	*IN*	*A_ARG_TYPE_DataTableID*
DataTransportURL	*OUT*	*A_ARG_TYPE_DataTransportURL*

5.7.9.2 Argument Descriptions

DataTableID: This argument shall provide an identifier associated with the target DataTable.

DataTransportURL: This argument shall return a URL which can support HTTP-POST requests to the DataTable identified by the *DataTableID* argument.

5.7.9.3 Service Requirements

If DeviceProtection is supported, this action shall only succeed if the invoking control point identity has sufficient access privileges. The DataStore service shall accept a HTTP-POST request to the URL provided by this action. The *request-body* of this HTTP-POST shall consist of a `<DataRecords>` element containing zero or more `<datarecord >`elements. The DataStore service shall respond to this HTTP-POST request with HTTP status 200. If an error is encountered during updating of the DataTable, the DataStore service shall respond to the HTTP-POST with a `<DataRecordsStatus>` element indicating which DataRecord(s) were successfully written. See subclause 5.3.10, "DataTable HTTP/HTTPS Transport Protocol" for further information.

5.7.9.4 Control Point Requirements When Calling The Action

None.

5.7.9.5 Dependency on Device State

None.

5.7.9.6 Effect on Device State

None.

5.7.9.7 Errors

Table 22 — Error Codes for *GetDataStoreTransportURL()*

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	DataTable not found	The DataTable indicated by DataTableID cannot be found.
703	Insufficient role(s) or permission(s).	The control point does not have sufficient roles to perform the requested action.
706	No transport URLs available.	All available TransportURLs for the indicated DataTable have been allocated.

ErrorCode	errorDescription	Description

5.7.10   *ModifyDataStoreTableInfo()*

This allowed action supports modification of selected elements of an existing DataTable indicated by the *DataTableID* argument.

5.7.10.1   Arguments

Table 23 — Arguments for *ModifyDataStoreTableInfo()*

Argument	Direction	relatedStateVariable
DataTableID	*IN*	*A_ARG_TYPE_DataTableID*
DataTableInfoElementOrig	*IN*	*A_ARG_TYPE_DataTableInfoFragment*
DataTableInfoElementNew	*IN*	*A_ARG_TYPE_DataTableInfoFragment*

5.7.10.2   Argument Descriptions

DataTableID: This argument shall provide an identifier associated with the target DataTable.

DataTableInfoElementOrig: This argument contain shall provide an XML fragment from the target DataTable's UPnP DataStore DataTableInfo XML document [17]. If a new element is being added to the target DataTable, then an empty top-level element shall be provided. The XML fragment provided shall start with one of the following DataTable XML elements:

- `<datatableretain>`
- `<datatableroles>`
- `<datatablegroups>`

The contents of this argument shall be consistent with the current DataTable state except as noted above for new elements.

DataTableInfoElementNew: This argument shall contain an updated copy of the XML fragment from to be replaced or added in the target DataTable.

5.7.10.3   Service Requirements

If *DeviceProtection* is supported, this action shall only succeed if the invoking control point identity has sufficient access privileges. A control point with an *Admin* role may modify any of the DataTable elements listed above. Otherwise, if the control point is modifying the `<datatablegroups>` or `<datatableretain>` elements, then the control point shall have *ds:Master#[GroupName]* roles for all existing groups the DataTable currently references and also *ds:Master#[GroupName]* roles for any additional groups to be added. If the control point is modifying or adding a `<datatablesroles>` element, then the control point shall either have an Admin role or shall have *ds:Master#[GroupName]* roles for all existing groups the DataTable currently references.

There are a number of semantic conditions that may cause an implementation to reject a DataTable modification request. These include: attempts to modify DataTable elements other than listed in the action description; a *DataTableInfoElementOrig* argument which is not consistent with the current DataTable state; a *DataTableInfoElementNew* argument missing elements specified by the *DataTableInfoElementOrig* argument. For modifications which are not rejected due to incorrect group or device protection issues, error code 714 shall be returned. However, it is recommended that implementation(s) provide additional supporting diagnostics to assist in identifying the causes of rejected modification requests.

5.7.10.4 Control Point Requirements When Calling The Action

None.

5.7.10.5 Dependency on Device State

None.

5.7.10.6 Effect on Device State

A LastChange event notification (updateType attribute values: "*G*" or "*O*") shall be generated due to this action. See LastChange state variable.

5.7.10.7 Errors

Table 24 — Error Codes for *ModifyDataStoreTableInfo()*

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	The (*DataTableInfoElementOrig* or *DataTableInfoElementNew*) argument XML elements are not valid.
702	DataTable not found	The DataTable indicated by DataTableID cannot be found.
703	Insufficient role(s) or permission(s).	The control point does not have sufficient roles to perform the requested action.
704	Invalid group(s),	The DataStore group(s) do not exist.
705	Invalid role(s) or permission(s).	The DataTable Role(s) or Permission(s) do not exist.
714	DataTable modification not acceptable	The DataTable modification requested is not acceptable due to semantic errors.

5.7.11 *ReadDataStoreTableRecords()*

This required action shall return DataRecord(s) from the DataTable indicated by the *DataTableID* argument. The *DataRecordFilter* argument may specify a string to indicate which DataRecord(s) shall be returned.

5.7.11.1 Arguments

Table 25 — Arguments for *ReadDataStoreTableRecords()*

Argument	Direction	relatedStateVariable
DataTableID	*IN*	*A_ARG_TYPE_DataTableID*
DataRecordFilter	*IN*	*A_ARG_TYPE_DataRecordFilter*
DataRecordStart	*IN*	*A_ARG_TYPE_DataRecordIndex*
DataRecordCount	*IN*	*A_ARG_TYPE_DataRecordCount*
DataRecordPropResolve	*IN*	*A_ARG_TYPE_DataRecordPropResolve*
DataRecords	*OUT*	*A_ARG_TYPE_DataRecords*
DataRecordContinue	*OUT*	*A_ARG_TYPE_DataRecordIndex*

5.7.11.2 Argument Descriptions

DataTableID: This argument shall provide an identifier associated with the target *DataTable*.

DataRecordFilter: This argument shall contain a string used to qualify *DataRecord(s)* to be returned by this action. This argument shall either be the empty string indicating no filter is to

be applied or a string containing an DataRecordFilter XML document [20]. See *A_ARG_TYPE_DataRecordFilter* for additional details.

DataRecordStart: This argument shall contain the starting point if multiple invocations of this action are required to return all matching *DataRecord(s)*. A value of ("*0*") indicates that the first available DataRecord shall be considered. See the *DataRecordContinue* argument for additional details

DataRecordCount: This argument shall indicate the maximum number of matching DataRecord(s) to be returned by this invocation. A value of ("*0*") indicates that no limit on the number of DataRecord(s) returned.

DataRecordPropResolve: If this argument is ("*1*") then the DataStore service shall attempt to resolve any DataRecord fields defined with the `tableprop` attribute value of ("*1*") by using the corresponding DataItem value as a DataTable Dictionary key. The returned DataTable record DataItem shall contain the corresponding DataTable Dictionary key's value. If the named DataTable Dictionary key is missing, the corresponding returned DataItem shall be empty. If this argument is ("*0*") then the unresolved DataTable key is returned is returned for the DataItem value.

DataRecords: This argument shall return a UPnP DataStore DataRecord XML document containing DataRecord(s).

DataRecordContinue: This argument provides a value to be provided by the *DataRecordStart* argument to continue reading DataRecord(s) from the point at which a prior *ReadDataStoreTableRecords()* action had stopped. The previous *ReadDataStoreTableRecords()* action may have stopped due to satisfying the number of records specified by the *DataRecordCount* argument or due to a lack of available DataRecord(s). Note: The meaning of *DataRecordStart* and *DataRecordContinue* argument values other than ("*0*") are implementation specific.

5.7.11.3   Service Requirements

If DeviceProtection is supported, this action shall only succeed if the invoking control point identity has sufficient access privileges.

When specifying *DataRecordStart* values based on a prior *DataRecordContinue* value, the implementation may determine that the DataRecord being referred to no longer exist in the DataTable due to DataRecord retention criteria. In this case error code 711 shall be returned.

5.7.11.4   Control Point Requirements When Calling The Action
None.

5.7.11.5   Dependency on Device State
None.

5.7.11.6   Effect on Device State
None.

5.7.11.7   Errors

Table 26 — Error Codes for *ReadDataStoreTableRecords()*

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	The (*DataRecordFilter*) argument XML elements are not valid.

ErrorCode	errorDescription	Description
702	DataTable not found	The DataTable indicated by DataTableID cannot be found.
703	Insufficient role(s) or permission(s).	The control point does not have sufficient roles to perform the requested action.
709	Invalid filter	The filter argument (DataRecordFilter) is not valid.
711	Invalid record index	The record index argument (DataRecordStart) is no longer valid.

5.7.12 *RemoveDataStoreTableKeyValue()*

This conditionally required action removes the indicated DataTable Dictionary Key. This action shall be implemented if the *SetDataStoreTableKeyValue()* action is supported.

5.7.12.1 Arguments

Table 27 — Arguments for *RemoveDataStoreTableKeyValue()*

Argument	Direction	relatedStateVariable
DataTableID	*IN*	*A_ARG_TYPE_DataTableID*
DataTableKeyName	*IN*	*A_ARG_TYPE_DataTableKeyName*

5.7.12.2 Argument Descriptions

DataTableID: This argument shall provide an identifier associated with the target DataTable.

DataTableKeyName: This argument identifies the *DataTable* Dictionary key to be removed.

5.7.12.3 Service Requirements

The DataStore service shall verify that the requesting control point identity has sufficient privileges to write the referenced DataTable.

5.7.12.4 Control Point Requirements When Calling The Action

A control point identity invoking this action shall have sufficient privileges to write the indicated DataTable.

5.7.12.5 Dependency on Device State

None.

5.7.12.6 Effect on Device State

A LastChange event notification (`updateType` attribute values: "*P*") shall be generated due to this action. See LastChange state variable.

5.7.12.7 Errors

Table 28 — Error Codes for *RemoveDataStoreTableKeyValue()*

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	DataTable not found	The DataTable indicated by DataTableID cannot be found.
703	Insufficient role(s) or permission(s).	The control point does not have sufficient roles to perform the requested action.
707	Key name not found	The indicated DataTable Dictionary KeyName was not found.

ErrorCode	errorDescription	Description
708	Key name invalid	The key name provided is invalid (ex: empty string)

5.7.13 *ResetDataStoreTable ()*

This required action supports clearing all DataRecord(s), DataTable Dictionary entries and Transport URLs for the DataTable indicated by the *DataTableID* argument.

5.7.13.1 Arguments

Table 29 — Arguments for *ResetDataStoreTable()*

Argument	Direction	relatedStateVariable
DataTableID	*IN*	*A_ARG_TYPE_DataTableID*
ResetDataTableRecords	*IN*	*A_ARG_TYPE_DataTableResetReq*
ResetDataTableDictionary	*IN*	*A_ARG_TYPE_DataTableResetReq*
ResetDataTableTransport	*IN*	*A_ARG_TYPE_DataTableResetReq*

5.7.13.2 Argument Descriptions

DataTableID: This argument shall provide an identifier associated with the target DataTable.

ResetDataTableRecords: If this argument is set to ("*1*"), all existing DataRecords are removed from the target DataTable.

ResetDataTableDictionary: If this argument is set to ("*1*"), all existing DataTable Dictionary entries are removed from the target DataTable.

ResetDataTableTransport: If this argument is set to ("*1*"), all existing DataTable data TransportURLs to the target DataTable are invalidated.

5.7.13.3 Service Requirements

The DataStore service shall verify that the requesting control point identity has *ds:Master* or *Admin* privileges for the referenced DataTable. If this action requests multiple DataTable operations, then the requested operations shall complete atomically with respect to DataStore clients. If a *ResetDataTableTransport* operation is requested, subsequent connection attempts to existing DataTable TransportURLs shall respond with HTTP status 410 "Gone".

5.7.13.4 Control Point Requirements When Calling The Action

A control point identity invoking this action shall have *ds:Master* or *Admin* privileges for the indicated DataTable.

5.7.13.5 Dependency on Device State

None.

5.7.13.6 Effect on Device State

A LastChange event notification (`updateType` attribute values: "*X*") shall be generated due to this action. See LastChange state variable.

5.7.13.7 Errors

Table 30 — Error Codes for *ResetDataStoreTable()*

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.

DataStore:1

ErrorCode	errorDescription	Description
702	DataTable not found	The DataTable indicated by DataTableID cannot be found.
703	Insufficient role(s) or permission(s).	The control point does not have sufficient roles to perform the requested action.

5.7.14 *SetDataStoreTableKeyValue()*

This conditionally allowed   action updates a DataTable Dictionary key for the indicated DataTable. If the Dictionary key does not exist it is created, otherwise the existing key value is updated. A DataStore implementation may only employ predefined DataStore tables in which case this action shall not be implemented.

5.7.14.1   Arguments

Table 31 — Arguments for *SetDataTableKeyValue()*

Argument	Direction	relatedStateVariable
DataTableID	*IN*	*A_ARG_TYPE_DataTableID*
DataTableKeyName	*IN*	*A_ARG_TYPE_DataTableKeyName*
DataTableKeyValue	*IN*	*A_ARG_TYPE_DataTableKeyValue*

5.7.14.2   Argument Descriptions

DataTableID: This argument shall provide an identifier associated with the target DataTable.

DataTableKeyName: This argument identifies the DataTable Dictionary key value to be set.

DataTableKeyValue: This argument provides the value of the identified DataTable Dictionary key value.

5.7.14.3   Service Requirements

The DataStore service shall verify that the requesting control point identity has sufficient privileges to write the referenced DataTable.

5.7.14.4   Control Point Requirements When Calling The Action

A control point identity invoking this action shall have sufficient privileges to write the indicated DataTable.

5.7.14.5   Dependency on Device State

None.

5.7.14.6   Effect on Device State

A LastChange event notification (`updateType` attribute values: "*P*") shall be generated due to this action.

5.7.14.7   Errors

Table 32 — Error Codes for *SetDataStoreTableKeyValue()*

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	DataTable not found	The DataTable indicated by DataTableID cannot be found.

DataStore:1

ErrorCode	errorDescription	Description
703	Insufficient role(s) or permission(s).	The control point does not have sufficient roles to perform the requested action.
708	Key name invalid	The key name provided is invalid (ex: empty string)

5.7.15 *WriteDataStoreTableRecords()*

This required action shall write DataRecord(s) to the DataTable identified by the *DataTableID* argument.

5.7.15.1 Arguments

Table 33 — Arguments for *WriteDataStoreTableRecords()*

Argument	Direction	relatedStateVariable
DataTableID	*IN*	*A_ARG_TYPE_DataTableID*
DataRecords	*IN*	*A_ARG_TYPE_DataRecords*
DataRecordsStatus	*OUT*	*A_ARG_TYPE_DataRecordsStatus*

5.7.15.2 Argument Descriptions

DataTableID: This argument shall provide an identifier associated with the target *DataTable*.

DataRecords: This argument shall provide a UPnP DataStore DataRecord XML document [19] containing one or more DataRecord(s) to be written to the target DataTable.

DataRecordsStatus: This argument shall provide a UPnP DataRecordsStatus XML document [21]. If the DataStore did not accept one or more DataRecord(s) contained in the *DataRecords* argument, this argument shall indicate which data records were not accepted. If all DataRecords were accepted, then this argument shall return ("").

5.7.15.3 Service Requirements

If *DeviceProtection* is supported, this action shall only succeed if the invoking control point identity has sufficient access privileges.

5.7.15.4 Control Point Requirements When Calling The Action

None.

5.7.15.5 Dependency on Device State

None.

5.7.15.6 Effect on Device State

A LastChange event notification (updateType attribute values: "R" shall be generated due to this action.

5.7.15.7 Errors

Table 34 — Error Codes for *WriteDataStoreTableRecords()*

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	The (*DataRecords*) argument XML elements are not valid.

ErrorCode	errorDescription	Description
702	DataTable not found	The DataTable indicated by DataTableID cannot be found.
703	Insufficient role(s) or permission(s).	The control point does not have sufficient roles to perform the requested action.
712	DataItem not found	A DataRecord to be written contains a DataItem not specified by this DataTable.
713	DataItem item missing	The DataItem to be written is missing a DataItem required by this DataTable.

5.7.16 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 35 — 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	The argument XML elements are not valid.
702	DataTable not found	The DataTable indicated cannot be found.
703	Insufficient role(s) or permission(s).	The control point does not have sufficient roles to perform the requested action.
704	Invalid group(s).	The DataStore group(s) are not acceptable based on the action. See action description.
705	Invalid role(s) or permission(s).	The DataTable Role(s) or Permission(s) do not exist.
706	No transport URLs available.	All available TransportURLs for the indicated DataTable have been allocated.
707	Key name not found	The indicated Dictionary key name was not found.
708	Key name invalid	The key name provided is invalid (ex: empty string)
709	Invalid filter	The record filter argument is not valid.
710	Groups in use	The DataStore group(s) cannot be modified or deleted since they are currently referenced by DataTable(s).
711	Invalid record index	The DataRecord indicated by the argument value no longer exists in the DataTable.
712	DataItem not found	A DataRecord to be written contains a DataItem not specified by this DataTable.
713	DataItem item missing	The DataItem to be written is missing a DataItem required by this DataTable.
714	DataTable modification not acceptable	The DataTable modification requested is not acceptable due to semantic errors.
715	Reserved group name.	The DataStore group name has been reserved for use by the implementation.

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

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

    <actionList>

        <action>
            <name>CreateDataStoreGroups</name>
            <argumentList>
                <argument>
                    <name>DataStoreGroupList</name>
                    <direction>in</direction>
                    <relatedStateVariable>
                        A_ARG_TYPE_DataStoreGroups
                    </relatedStateVariable>
                </argument>
            </argumentList>
        </action>
        <action>
            <name>CreateDataStoreTable</name>
            <argumentList>
                <argument>
                    <name>DataTableInfo</name>
                    <direction>in</direction>
                    <relatedStateVariable>
                        A_ARG_TYPE_DataTableInfo
                    </relatedStateVariable>
                </argument>
                <argument>
                    <name>DataTableID</name>
                    <direction>out</direction>
                    <relatedStateVariable>
                        A_ARG_TYPE_DataTableID
                    </relatedStateVariable>
                </argument>
            </argumentList>
        </action>
        <action>
            <name>DeleteDataStoreGroups</name>
            <argumentList>
                <argument>
                    <name>DataStoreGroupList</name>
                    <direction>in</direction>
                    <relatedStateVariable>
                        A_ARG_TYPE_DataStoreGroups
                    </relatedStateVariable>
                </argument>
            </argumentList>
        </action>
        <action>
            <name>DeleteDataStoreTable</name>
            <argumentList>
                <argument>
                    <name>DataTableID</name>
                    <direction>in</direction>
                    <relatedStateVariable>
                        A_ARG_TYPE_DataTableID
                    </relatedStateVariable>
                </argument>
            </argumentList>
        </action>
        <action>
```

```xml
        <name>GetDataStoreTableKeyValue</name>
        <argumentList>
            <argument>
                <name>DataTableID</name>
                <direction>in</direction>
                <relatedStateVariable>
                    A_ARG_TYPE_DataTableID
                </relatedStateVariable>
            </argument>
            <argument>
                <name>DataTableKeyName</name>
                <direction>in</direction>
                <relatedStateVariable>
                    A_ARG_TYPE_DataTableKeyName
                </relatedStateVariable>
            </argument>
            <argument>
                <name>DataTableKeyValue</name>
                <direction>out</direction>
                <relatedStateVariable>
                    A_ARG_TYPE_DataTableKeyValue
                </relatedStateVariable>
            </argument>
        </argumentList>
    </action>
    <action>
        <name>GetDataStoreGroups</name>
        <argumentList>
            <argument>
                <name>DataStoreGroupList</name>
                <direction>out</direction>
                <relatedStateVariable>
                    A_ARG_TYPE_DataStoreGroups
                </relatedStateVariable>
            </argument>
        </argumentList>
    </action>
    <action>
        <name>GetDataStoreInfo</name>
        <argumentList>
            <argument>
                <name>DataStoreInfo</name>
                <direction>out</direction>
                <relatedStateVariable>
                    A_ARG_TYPE_DataStoreInfo
                </relatedStateVariable>
            </argument>
        </argumentList>
    </action>
    <action>
        <name>GetDataStoreTableInfo</name>
        <argumentList>
            <argument>
                <name>DataTableID</name>
                <direction>in</direction>
                <relatedStateVariable>
                    A_ARG_TYPE_DataTableID
                </relatedStateVariable>
            </argument>
            <argument>
                <name>DataTableInfo</name>
                <direction>out</direction>
                <relatedStateVariable>
                    A_ARG_TYPE_DataTableInfo
                </relatedStateVariable>
            </argument>
        </argumentList>
    </action>
```

DataStore:1

```xml
<action>
    <name>GetDataStoreTransportURL</name>
    <argumentList>
        <argument>
            <name>DataTableID</name>
            <direction>in</direction>
            <relatedStateVariable>
                A_ARG_TYPE_DataTableID
            </relatedStateVariable>
        </argument>
        <argument>
            <name>DataTransportURL</name>
            <direction>out</direction>
            <relatedStateVariable>
                A_ARG_TYPE_DataTransportURL
            </relatedStateVariable>
        </argument>
    </argumentList>
</action>
<action>
    <name>ModifyDataStoreTable</name>
    <argumentList>
        <argument>
            <name>DataTableID</name>
            <direction>in</direction>
            <relatedStateVariable>
                A_ARG_TYPE_DataTableID
            </relatedStateVariable>
        </argument>
        <argument>
            <name>DataTableInfoElementOrig</name>
            <direction>in</direction>
            <relatedStateVariable>
                A_ARG_TYPE_DataTableInfoElement
            </relatedStateVariable>
        </argument>
        <argument>
            <name>DataTableInfoElementNew</name>
            <direction>in</direction>
            <relatedStateVariable>
                A_ARG_TYPE_DataTableInfoElement
            </relatedStateVariable>
        </argument>
    </argumentList>
</action>
<action>
    <name>ReadDataStoreTableRecords</name>
    <argumentList>
        <argument>
            <name>DataTableID</name>
            <direction>in</direction>
            <relatedStateVariable>
                A_ARG_TYPE_DataTableID
            </relatedStateVariable>
        </argument>
        <argument>
            <name>DataRecordFilter</name>
            <direction>in</direction>
            <relatedStateVariable>
                A_ARG_TYPE_DataRecordFilter
            </relatedStateVariable>
        </argument>
        <argument>
            <name>DataRecordStart</name>
            <direction>in</direction>
            <relatedStateVariable>
                A_ARG_TYPE_DataRecordIndex
            </relatedStateVariable>
        </argument>
```

```xml
                        </argument>
                        <argument>
                            <name>DataRecordCount</name>
                            <direction>in</direction>
                            <relatedStateVariable>
                                A_ARG_TYPE_DataRecordCount
                            </relatedStateVariable>
                        </argument>
                        <argument>
                            <name>DataRecordPropResolve</name>
                            <direction>in</direction>
                            <relatedStateVariable>
                                A_ARG_TYPE_DataRecordPropResolve
                            </relatedStateVariable>
                        </argument>
                        <argument>
                            <name>DataRecords</name>
                            <direction>out</direction>
                            <relatedStateVariable>
                                A_ARG_TYPE_DataRecords
                            </relatedStateVariable>
                        </argument>
                        <argument>
                            <name>DataRecordContinue</name>
                            <direction>out</direction>
                            <relatedStateVariable>
                                A_ARG_TYPE_DataRecordIndex
                            </relatedStateVariable>
                        </argument>
                    </argumentList>
                </action>
                <action>
                    <name>RemoveDataStoreTableKeyValue</name>
                    <argumentList>
                        <argument>
                            <name>DataTableID</name>
                            <direction>in</direction>
                            <relatedStateVariable>
                                A_ARG_TYPE_DataTableID
                            </relatedStateVariable>
                        </argument>
                        <argument>
                            <name>DataTableKeyName</name>
                            <direction>in</direction>
                            <relatedStateVariable>
                                A_ARG_TYPE_DataTableKeyName
                            </relatedStateVariable>
                        </argument>
                    </argumentList>
                </action>
                <action>
                    <name>ResetDataStoreTable</name>
                    <argumentList>
                        <argument>
                            <name>DataTableID</name>
                            <direction>in</direction>
                            <relatedStateVariable>
                                A_ARG_TYPE_DataTableID
                            </relatedStateVariable>
                        </argument>
                        <argument>
                            <name>ResetDataTableRecords</name>
                            <direction>in</direction>
                            <relatedStateVariable>
                                A_ARG_TYPE_DataTableResetReq
                            </relatedStateVariable>
                        </argument>
                        <argument>
```

```xml
                    <name>ResetDataTableDictionary</name>
                    <direction>in</direction>
                    <relatedStateVariable>
                        A_ARG_TYPE_DataTableResetReq
                    </relatedStateVariable>
                </argument>
                <argument>
                    <name>ResetDataTableTransport</name>
                    <direction>in</direction>
                    <relatedStateVariable>
                        A_ARG_TYPE_DataTableResetReq
                    </relatedStateVariable>
                </argument>
            </argumentList>
        </action>
        <action>
            <name>SetDataStoreTableKeyValue</name>
            <argumentList>
                <argument>
                    <name>DataTableID</name>
                    <direction>in</direction>
                    <relatedStateVariable>
                        A_ARG_TYPE_DataTableID
                    </relatedStateVariable>
                </argument>
                <argument>
                    <name>DataTableKeyName</name>
                    <direction>in</direction>
                    <relatedStateVariable>
                        A_ARG_TYPE_DataTableKeyName
                    </relatedStateVariable>
                </argument>
                <argument>
                    <name>DataTableKeyValue</name>
                    <direction>in</direction>
                    <relatedStateVariable>
                        A_ARG_TYPE_DataTableKeyValue
                    </relatedStateVariable>
                </argument>
            </argumentList>
        </action>
        <action>
            <name>WriteDataStoreTableRecords</name>
            <argumentList>
                <argument>
                    <name>DataTableID</name>
                    <direction>in</direction>
                    <relatedStateVariable>
                        A_ARG_TYPE_DataTableID
                    </relatedStateVariable>
                </argument>
                <argument>
                    <name>DataRecords</name>
                    <direction>in</direction>
                    <relatedStateVariable>
                        A_ARG_TYPE_DataRecords
                    </relatedStateVariable>
                </argument>
                <argument>
                    <name>DataRecordsStatus</name>
                    <direction>out</direction>
                    <relatedStateVariable>
                        A_ARG_TYPE_DataRecordsStatus
                    </relatedStateVariable>
                </argument>
            </argumentList>
        </action>
    </actionList>
```

```xml
<serviceStateTable>
    <stateVariable sendEvents="yes">
        <name>LastChange</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_DataRecordIndex</name>
        <dataType>string</dataType>
    </stateVariable>
    <stateVariable sendEvents="no">
        <name>A_ARG_TYPE_DataRecordFilter</name>
        <dataType>string</dataType>
    </stateVariable>
    <stateVariable sendEvents="no">
        <name>A_ARG_TYPE_DataTableID</name>
        <dataType>string</dataType>
    </stateVariable>
    <stateVariable sendEvents="no">
        <name>A_ARG_TYPE_DataRecordPropResolve</name>
        <dataType>boolean</dataType>
    </stateVariable>
    <stateVariable sendEvents="no">
        <name>A_ARG_TYPE_DataTableInfoFragment</name>
        <dataType>string</dataType>
    </stateVariable>
    <stateVariable sendEvents="no">
        <name>A_ARG_TYPE_DataTableKeyName</name>
        <dataType>string</dataType>
    </stateVariable>
    <stateVariable sendEvents="no">
        <name>A_ARG_TYPE_DataTableKeyValue</name>
        <dataType>string</dataType>
    </stateVariable>
    <stateVariable sendEvents="no">
        <name>A_ARG_TYPE_DataTableInfo</name>
        <dataType>string</dataType>
    </stateVariable>
    <stateVariable sendEvents="no">
        <name>A_ARG_TYPE_DataTableResetReq</name>
        <dataType>boolean</dataType>
    </stateVariable>
    <stateVariable sendEvents="no">
        <name>A_ARG_TYPE_DataStoreGroups</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_DataRecordsStatus</name>
        <dataType>string</dataType>
    </stateVariable>
    <stateVariable sendEvents="no">
        <name>A_ARG_TYPE_DataTransportURL</name>
        <dataType>string</dataType>
    </stateVariable>
</serviceStateTable>
</scpd>
```