Open Group Standard

Open Data Format (O-DF),
an Open Group Internet of Things (IoT) Standard

Copyright © 2014-2017, The Open Group

The Open Group hereby authorizes you to use this document for any purpose, PROVIDED THAT any copy of this document, or any part thereof, which you make shall retain all copyright and other proprietary notices contained herein.

This document may contain other proprietary notices and copyright information.

Nothing contained herein shall be construed as conferring by implication, estoppel, or otherwise any license or right under any patent or trademark of The Open Group or any third party. Except as expressly provided above, nothing contained herein shall be construed as conferring any license or right under any copyright of The Open Group.

Note that any product, process, or technology in this document may be the subject of other intellectual property rights reserved by The Open Group, and may not be licensed hereunder.

This document is provided "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. Some jurisdictions do not allow the exclusion of implied warranties, so the above exclusion may not apply to you.

Any publication of The Open Group may include technical inaccuracies or typographical errors. Changes may be periodically made to these publications; these changes will be incorporated in new editions of these publications. The Open Group may make improvements and/or changes in the products and/or the programs described in these publications at any time without notice.

Should any viewer of this document respond with information including feedback data, such as questions, comments, suggestions, or the like regarding the content of this document, such information shall be deemed to be non-confidential and The Open Group shall have no obligation of any kind with respect to such information and shall be free to reproduce, use, disclose, and distribute the information to others without limitation. Further, The Open Group shall be free to use any ideas, concepts, know-how, or techniques contained in such information for any purpose whatsoever including but not limited to developing, manufacturing, and marketing products incorporating such information.

If you did not obtain this copy through The Open Group, it may not be the latest version. For your convenience, the latest version of this publication may be downloaded at www.opengroup.org/bookstore.

Open Group Standard

Open Data Format (O-DF), an Open Group Internet of Things (IoT) Standard

ISBN: 1-937218-59-1

Document Number: C14A

Published by The Open Group, October 2014.
Updated in September 2017 to apply Technical Corrigendum No. 1 (U173), which updates Appendix A.

Comments relating to the material contained in this document may be submitted to:

The Open Group, Apex Plaza, Forbury Road, Reading, Berkshire, RG1 1AX, United Kingdom

or by electronic mail to:

ogspecs@opengroup.org

Contents

List of Figures

List of Examples

Preface

The Open Group

The Open Group is a global consortium that enables the achievement of business objectives through IT standards. With more than 400 member organizations, The Open Group has a diverse membership that spans all sectors of the IT community – customers, systems and solutions suppliers, tool vendors, integrators, and consultants, as well as academics and researchers – to:

- Capture, understand, and address current and emerging requirements, establish policies, and share best practices
- Facilitate interoperability, develop consensus, and evolve and integrate specifications and open source technologies
- Operate the industry's premier certification service

Further information on The Open Group is available at www.opengroup.org.

The Open Group publishes a wide range of technical documentation, most of which is focused on development of Open Group Standards and Guides, but which also includes white papers, technical studies, certification and testing documentation, and business titles. Full details and a catalog are available at www.opengroup.org/bookstore.

Readers should note that updates – in the form of Corrigenda – may apply to any publication. This information is published at www.opengroup.org/corrigenda.

This Document

This document specifies the Open Data Format (O-DF), an Open Group Internet of Things (IoT) Standard. It has been developed and approved by The Open Group.

This document is structured as follows:

- Chapter 1 provides an introduction to the standard and describes conformance requirements, normative references, and terminology
- Chapter 2 describes the O-DF elements and attributes and how they are expected to be used
- Chapter 3 describes how the O-DF can be used directly in RESTful services
- Chapter 4 explains how to create domain-specific data models that are extensions of the O-DF
- Appendix A contains the XML Schema file
- Appendix B contains some example O-DF structures
- Appendix C shows a possible JSON mapping

Trademarks

TOGAF®, UNIX®, UNIXWARE®, X/Open®, and the Open Brand X® logo are registered trademarks and Boundaryless Information Flow™, Build with Integrity Buy with Confidence™, Dependability Through Assuredness™, EMMM™, FACE™, the FACE™ logo, IT4IT™, the IT4IT™ logo, O-DEF™, O-PAS™, Open FAIR™, Open Platform 3.0™, Open Process Automation™, Open Trusted Technology Provider™, Platform 3.0™, SOSA™, the Open O™ logo, and The Open Group Certification logo (Open O and check™) are trademarks of The Open Group.

Java® is a registered trademark of Oracle and/or its affiliates.

All other brands, company, and product names are used for identification purposes only and may be trademarks that are the sole property of their respective owners.

Acknowledgements

The Open Group gratefully acknowledges the contribution of the following people in the development of this standard.

Contributing Members of The Open Group QLM Work Group

Organization	Contributor(s)
Aalto University (former Helsinki University of Technology)	Kary Främling* (QLM Work Group Chair) Andrea Buda, Sylvain Kubler, Manik Madhikermi, Merina Maharjan
BIBA	Johannes Lützenberger
EPFL	Dimitris Kiritsis, Nikolaos Maris, Min-Jung Yoo
Holonix Srl	Jacopo Cassina, Eva Coscia, Simone Parrotta
Promise Innovation	David Potter
UDEF-IT, LLC	Ron Schuldt, Roberta Shauger

* Primary Contributor

Open Group Staff

Name	Role
Martin Kirk	Director, The Open Group QLM Work Group (until May 2014)
Chris Harding	Director, The Open Group QLM Work Group (since June 2014)

PMI

This O-DF specification is to a great extent based on the earlier PROMISE Messaging Interface (PMI), developed during the PROMISE EU FP7 project.

Organization	Contributor(s)
Aalto University (former Helsinki University of Technology)	Kary Främling*, Vincent Michel, Jan Nyman
BIBA	Celal Dikici, Karl Hribernik, Robertino Solanas
Cambridge University	James Brusey, Mark Harrison, Duncan McFarlane

Organization	Contributor(s)
Infineon	Daniel Barisic, Guido Stromberg
InMediasP GmbH	Andreas Edler, Michael Marquard
Politecnico di Milano	Jacopo Cassina
Promise Innovation	David Potter
SAP	Jürgen Anke, Falk Brauer, Gregor Hackenbroich, Mario Neugebauer
Trackway Ltd.	Björn Forss*, Jouni Petrow

* Primary Contributor

Referenced Documents

Normative References

Normative references for this standard are defined in Section 1.4.

Informative References

The following documents are referenced in this standard:

- An Introduction to Quantum Lifecycle Management (QLM): Maximizing Business Value through Whole-of-Life Lifecycle Management, White Paper (W12A), November 2012, published by The Open Group; refer to: www.opengroup.org/bookstore/catalog/w12a.htm
- Open Group Internet of Things (IoT) Standard: Open Messaging Interface (O-MI), (C14B), October 2014, published by The Open Group; refer to: www.opengroup.org/bookstore/catalog/c14b.htm

1 Introduction

1.1 Objective

The Open Group Internet of Things (IoT) Standards have been developed to fill an interoperability gap identified in the context of the IoT, as explained in the White Paper: An Introduction to Quantum Lifecycle Management (QLM).

A great number of useful standards exist on the level of communications within a local network, within a specific domain, or for a limited purpose such as remote management of computers. However, at the moment of writing this specification, we have not been able to identify an appropriate standard that would address the higher-level requirements of the IoT. Such requirements are notably the need for any data sources (devices, machines, server-based systems, etc.) to be able to publish their available data and provide access to it in an easy and secure way, which includes the possibility to filter the data provided depending on the requester's identity, the context, etc.

The O-DF can be used for publishing the available data using ordinary URL (Uniform Resource Locator) addresses. O-DF structures can also be used for requesting and sending published data between systems, notably when used together with the O-MI standard.

In the IoT, information about a product or a "Thing" is often distributed over many different devices, systems, and organizations. The O-DF is intended to represent information about things in a standardized way that can be understood and *exchanged* in a universal way by all information systems that need to manage such IoT-related data. A data format structure typically does not contain complete information about a particular thing. Information about the same thing may be contained in several different data format structures. Object identifiers make it possible to link the data about a single thing that may be located in different information systems. An object identifier may be the only information that a data format structure contains about a particular thing.

The visibility and the access to data may depend on the object identifier used, as well as on the identity of the requesting party, as well as on the context of the request. This is why the object identifier data structure is of particular importance in any universal IoT standard.

1.2 Overview

The O-DF is specified using XML Schema. It defines a simple and extensible ontology that allows the creation of information structures that are similar to those of objects and properties in object-oriented programming. It is thereby generic enough for the representation of any object and information that is needed for information exchange in domains such as the IoT, lifecycle information management, etc.

An O-DF structure is a hierarchy with an *Objects* element as its top element, as illustrated in Figure 1. The *Objects* element can contain any number of *Object* sub-elements. *Object* elements are identified by at least one *id* sub-element. An *Object* may also have an optional *description*

sub-element. *Object* elements usually have properties, which are sub-elements called *InfoItem*, as well as *Object* sub-elements. The resulting *Object* tree can contain any number of levels. Chapter 2 provides detailed, normative descriptions of these elements.

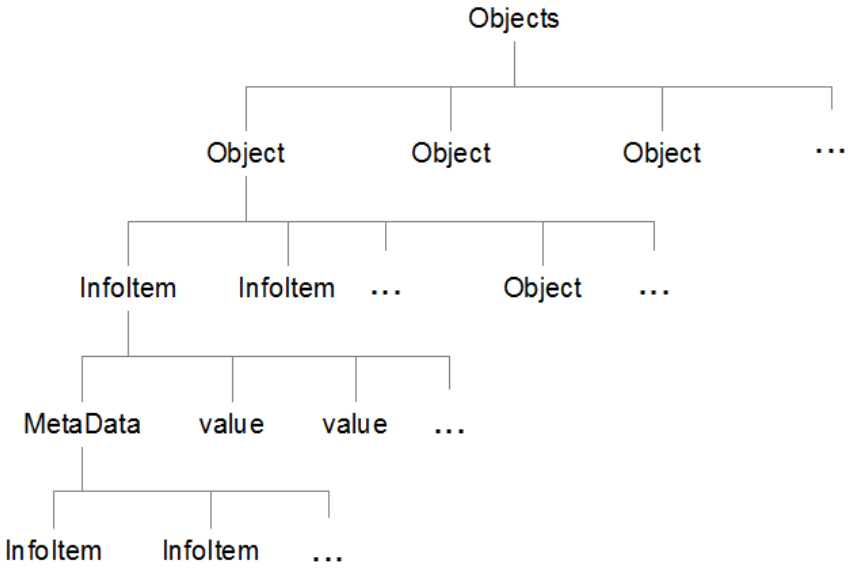

Figure 1: Illustration of O-DF Element Hierarchy

The O-DF is intended to be used for expressing information about "any" identifiable object (products, services, humans, …). How the information is communicated is not a part of this standard. The communication media might be a file sent as an email attachment, on a USB stick, or any other kind of media. O-DF content can also be sent using REST-based services, SOAP, Java Message Service (JMS), the O-MI, and other kinds of messaging protocols. The O-DF can be used as a query and response format in such messaging; for instance, the O-MI specifies that a "*read*" request with an O-DF structure should be responded to with the next level in the hierarchy shown in Figure 1. As an example, a request with only an "*Objects*" element should return an O-DF response with the list of *Object* elements available, including at least the compulsory attributes and sub-elements (notably at least one *id* element).

1.3 Conformance

This standard specifies conformance requirements for the O-DF and its specification and use. Chapters 2, 3, and 4 and Appendix A are normative. Appendices B and C are informative.

1.4 Normative References

The following standards contain provisions which, through references in this standard, constitute provisions of the O-DF standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards.

- XML Schema Part 2: Datatypes Second Edition, Ed. Biron & Malhotra; refer to: www.w3.org/TR/xmlschema-2.

1.5 Terminology

For the purposes of this standard, the following terminology definitions apply. When used in this way, these terms will always be shown in ALL CAPS; when these words appear in ordinary typeface they are intended to have their ordinary English meaning.

Can	Describes a permissible optional feature or behavior available to the user or application. The feature or behavior is mandatory for an implementation that conforms to this document. An application can rely on the existence of the feature or behavior.
May	Describes a feature or behavior that is optional for an implementation that conforms to this document. An application should not rely on the existence of the feature or behavior. An application that relies on such a feature or behavior cannot be assured to be portable across conforming implementations. To avoid ambiguity, the opposite of "may" is expressed as "need not", instead of "may not".
Must	Describes a feature or behavior that is mandatory for an application or user. An implementation that conforms to this document shall support this feature or behavior.
Shall	Describes a feature or behavior that is mandatory for an implementation that conforms to this document. An application can rely on the existence of the feature or behavior.
Should	For an implementation that conforms to this document, describes a feature or behavior that is recommended but not mandatory. An application should not rely on the existence of the feature or behavior. An application that relies on such a feature or behavior cannot be assured to be portable across conforming implementations. For an application, describes a feature or behavior that is recommended programming practice for optimum portability.
Will	Same meaning as "shall"; "shall" is the preferred term.

1.6 Future Directions

The Open Group IoT Standards will continue to be maintained by the QLM Work Group of The Open Group Platform 3.0™ Forum. They may be revised to correct errors or to incorporate changes based on implementation experience.

2 O-DF Objects

The XML Schema “odf.xsd” provides a formal specification for the O-DF. The schema contains annotations for Attributes and Child Elements that are identical to the ones in this standard. If there is any conflicting information between the schema file and this standard, then the information in the schema file is to be used.

The following units or formats SHALL be used in the O-DF to express the listed quantities. The *xs* namespace is defined in XML Schema Part 2: Datatypes Second Edition.

Table 1: O-DF Units and Formats

Value	Unit/Format
Date	Use xs:Date
Time	Use xs:dateTime
Duration	Use xs:duration However, it is defined in the schema where xs:duration should be used. For other durations, such as Time-To-Live (TTL), xs:double may be used and then the unit is seconds.
Other Values	SI units used by default.

As shown in Figure 1, all O-DF structures SHALL have *Objects* as their root element. An *Objects* element SHALL only have *Object* sub-elements. The most important attribute of an *Object* is *type*, which specifies what kind of object it is. An attribute MAY be used for specifying the object class. If an object class taxonomy is used, then it SHALL be indicated with the type attribute as a URL that points to the definition of the type; e.g., such as http://www.somewhere.com/taxonomy#theType. Example 1 shows a simple example of an O-DF structure.

Every *Object* SHALL have at least one sub-element called *id* that identifies the *Object*. An Object MAY have additional *id* elements. Such an identifier is known as an *object identifier*. An *object identifier* SHOULD be globally unique or at least unique for the specific application, domain, or network of the organizations involved. An optional *description* sub-element MAY also be included for providing a description of the *Object*, usually intended for human users.

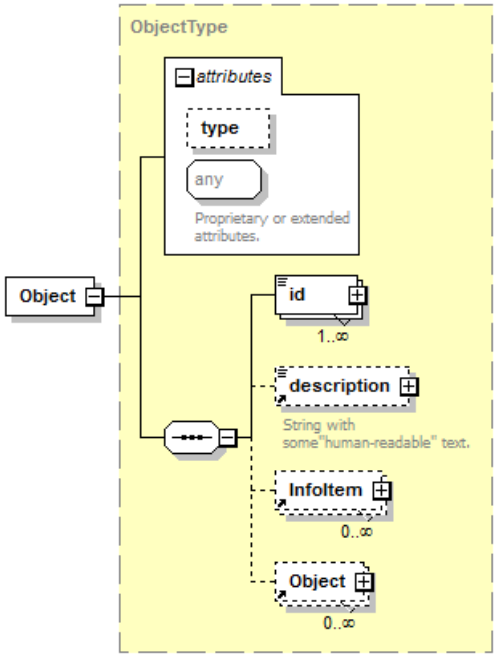

Figure 2: Object Element

The *Object*'s properties are included with an arbitrary number of *InfoItem* elements. *Object* elements MAY also include an arbitrary number of *Object* sub-elements (see Figure 2). The reason for calling properties "InfoItem", rather than using the name "Property" or some similar concept that is familiar from object-oriented programming, is that in the IoT an *InfoItem* can also be an event of some kind rather than just a simple value. The *name* attribute SHALL be used for defining the name of the *InfoItem*. Additional names – that is, synonyms or alternative names in different systems – MAY be provided as *name* sub-elements. As for *Object*, an attribute MAY also be used for specifying what the *InfoItem* represents.

InfoItem elements MAY contain optional sub-elements, as shown in Figure 3:

- *name*: additional names for the same *InfoItem*

  This feature may be used, for instance, if the same *InfoItem* is known under different names in different organizations or software.

- *description*: text that explains what the *InfoItem* represents, mainly intended for human users

- *MetaData*: sub-element that provides metadata information about the *InfoItem*, such as value type, units, and other similar information

- *value*: arbitrary number of values for the *InfoItem*, possibly with timestamps (Figure 4)

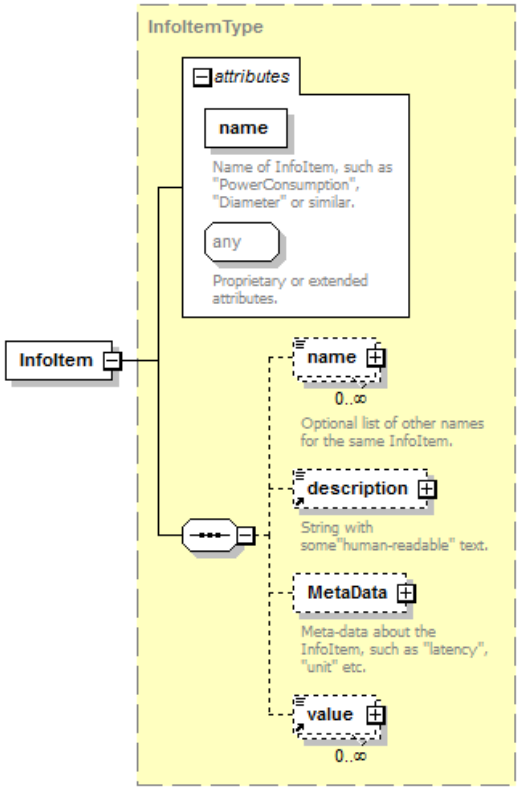

Figure 3: InfoItem Element

Even though it is possible to include *description*, *MetaData*, and *value* element(s) simultaneously for an *InfoItem*, it is usually not practical to do so. Metadata is typically requested only once when encountering a previously unknown *InfoItem*. The *MetaData* element MAY contain an arbitrary number of *InfoItem* elements. *MetaData* sub-elements are of *InfoItem* type because they are syntactically similar to *Object InfoItem* sub-elements, even though *MetaData InfoItems* are conceptually different from *Object InfoItems*. The *description* element could also be considered as metadata. However, it has been left as a separate element mainly due to earlier experience that has shown the utility of including a simple-to-use "free-form" text element for user interface and debugging purposes.

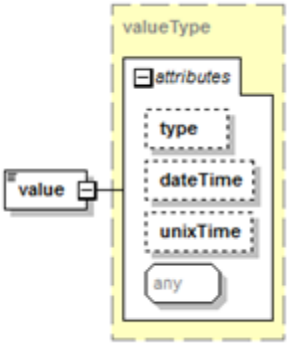

Figure 4: Value Element

Example 1 shows an O-DF structure that transmits two electrical power consumption measurements with timestamps for a refrigerator instance.

Example 1: O-DF Structure

```
<Objects>
  <Object type="Refrigerator Assembly Product">
    <id>SmartFridge22334411</id>
    <InfoItem name="Consumed Electrical Power Measure">
      <description>Power consumption values with timestamp.</ description >
      <value dateTime="2001-10-26T15:33:21">15.5</value>
      <value dateTime="2001-10-26T15:33:50">15.7</value>
    </InfoItem>
  </Object>
</Objects>
```

In object-oriented programming, objects are aware of each other by object containment hierarchies as illustrated in Figure 1 and by reference or pointers. In the O-DF partial object descriptions in different structures are linked using the *Object id* sub-element (Figure 2). In the IoT the *id* does not refer to a specific memory location but to an IoT object whose information may be spread over several information systems and organizations. Different methods and systems have been proposed for the discovery of such distributed information. The simplest mechanism is to include a URL in the *id* itself. Other methods are still being developed for solving this issue. However, those are not in the scope of this standard.

Object identifiers and *InfoItem* names are specified using the *qlmID* type (Figure 5). The attributes of *qlmID* make it possible to express what standard or coding scheme the *id*/*name* uses, on what kind of media it is written (e.g., RFID tag, barcode, stamped, …), and the beginning and end of validity of the identifier. Other information CAN be conveyed using attributes that are not defined in this standard. Especially for objects, the possibility to use more than one *id* is a common real-life requirement because the same "Thing" can often carry several different identifiers. For instance, a postal package may end up with several company-specific tracking numbers, an SSCC code, and other identifiers before reaching its destination.

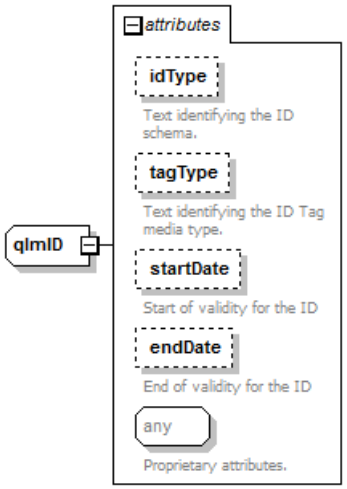

Figure 5: *qlmID* Type

3 RESTful Use of the O-DF

Due to the hierarchical nature of O-DF structures, they CAN be used for performing RESTful, URL-based information discovery and queries. An example of how this can be done using the UNIX "wget" utility is shown in Example 2, with a corresponding example response in Example 3. The response SHALL contain all compulsory attributes and sub-elements. It MAY also include other attributes and sub-elements.

Example 2: Issuing an HTTP GET Request

```
wget http://dialog.hut.fi/qlm/Objects/
```

Example 2 illustrates issuing an HTTP GET Request to the URL "http://dialog.hut.fi/qlm/"[1] for querying the available information about the data source "Objects".

Example 3: Response using the O-DF

```
<Objects>
    <Object>
        <id>Refrigerator123</id>
    </Object>
    <Object>
        <id>HeatingController321</id>
    </Object>
    <Object>
        <id>WeatherStation651</id>
    </Object>
</Objects>
```

Example 3 shows an example response using the O-DF to request an Objects list, within the "Smart Home" domain.

The elements of the retrieved O-DF structure in Example 3 can be used for drilling further down into the object hierarchy where, for instance, the following URL:

```
http://dialog.hut.fi/qlm/Objects/Refrigerator123/
```

would return the list of *InfoItem* sub-elements and possible sub-objects of the object "Refrigerator123" as shown in Example 4.

Example 4: Result for a URL-Based Data Discovery Request using O-DF Semantics

```
<Object>
    <id>Refrigerator123</id>
    <InfoItem name="PowerConsumption"/>
    <InfoItem name="RefrigeratorDoorOpenWarning"/>
    <InfoItem name="RefrigeratorProbeFault/>
</Object>
```

[1] The URL "http://dialog.hut.fi/qlm/" is provided as an example of a valid URL of an O-MI node. The reader should not assume that a valid O-MI node would be continuously available at that address, nor that it would return the content shown in this standard.

Further drilling down in the O-DF structure (Example 2) can be done in similar ways, as for example:

- `<URL>/Objects/Refrigerator123/id/` returns all the *id*s of Refrigerator123
- `<URL>/Objects/Refrigerator123/PowerConsumption/` returns the current power consumption value structure
- `<URL>/Objects/Refrigerator123/PowerConsumption/value/` returns the "raw" power consumption value
- `<URL>/qlm/Objects/Refrigerator123/PowerConsumption/MetaData/` returns the *MetaData* structure that corresponds to Refrigerator123's PowerConsumption
- `<URL>/qlm/Objects/Refrigerator123/PowerConsumption/name/` returns the PowerConsumption element including all its *name* sub-elements
- etc.

More complex queries such as querying for values of several *Objects* and *InfoItems* in one go, or querying for historical values, requires the use of other kinds of querying mechanisms, such as those specified in the O-MI.

4 Inheritance Mechanism for Domain-Specific Data Models

At the time of publication, a domain-specific data model extension exists only for product and product lifecycle-related information. That data model is specified in a separate XML Schema file. Such domain-specific schema SHALL include the O-DF schema by the following line:

```
"<xs:include schemaLocation="odf.xsd"/>".
```

The following lines define that a new type called *omiPhysicalProduct* is an extension of *Object* and can be used in the same way as *Object*:

```
<xs:element name="omiPhysicalProduct" type="PhysicalProduct"
substitutionGroup="Object"/>
<xs:complexType name="PhysicalProduct">
    <xs:complexContent>
    <xs:extension base="ObjectType">
…
```

This signifies that *omiPhysicalProduct* elements can be used interchangeably with *Object* elements and that they inherit all the attributes and sub-elements of *Object*. Other attributes and sub-elements can then be defined for the *PhysicalProduct* type, which are particular for that domain. Similar extensions can be created for all other data types defined in the root data model schema.

The same extension mechanism SHALL be used for all other O-DF-compliant specifications. It is currently foreseen that such specifications will be defined at least for the healthcare domain.

A O-DF XSD Schema (Normative)

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.opengroup.org/xsd/odf/1.0/"
targetNamespace="http://www.opengroup.org/xsd/odf/1.0/"
elementFormDefault="qualified" attributeFormDefault="unqualified"
version="1.0">
    <xs:element name="Objects" type="ObjectsType">
        <xs:annotation>
            <xs:documentation>Data Model Root Element</xs:documentation>
        </xs:annotation>
    </xs:element>
    <xs:complexType name="ObjectsType">
        <xs:sequence>
            <xs:element name="Object" type="ObjectType" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="version" type="xs:string" use="optional">
            <xs:annotation>
                <xs:documentation>Schema version used.</xs:documentation>
            </xs:annotation>
        </xs:attribute>
    </xs:complexType>
    <xs:complexType name="ObjectType">
        <xs:sequence>
            <xs:element name="id" type="QlmIDType" minOccurs="1"
maxOccurs="unbounded"/>
            <xs:element name="description" type="DescriptionType" minOccurs="0"
maxOccurs="unbounded"/>
            <xs:element name="InfoItem" type="InfoItemType" minOccurs="0"
maxOccurs="unbounded"/>
            <xs:element name="Object" type="ObjectType" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="type" type="xs:string" use="optional"/>
        <xs:anyAttribute processContents="lax">
            <xs:annotation>
                <xs:documentation>Proprietary or extended
attributes.</xs:documentation>
            </xs:annotation>
        </xs:anyAttribute>
    </xs:complexType>
    <xs:complexType name="InfoItemType">
        <xs:sequence>
            <xs:element name="name" type="QlmIDType" minOccurs="0"
maxOccurs="unbounded">
                <xs:annotation>
                    <xs:documentation>Optional list of other names for the same
InfoItem.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="description" type="DescriptionType" minOccurs="0"
maxOccurs="unbounded"/>
            <xs:element name="MetaData" type="MetaDataType" minOccurs="0"
maxOccurs="unbounded">
                <xs:annotation>
```

```
                    <xs:documentation>Meta-data about the InfoItem, such as
"latency", "unit" etc.</xs:documentation>
                </xs:annotation>
            </xs:element>
            <xs:element name="value" type="ValueType" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="name" type="xs:string" use="required">
            <xs:annotation>
                <xs:documentation>Name of InfoItem, such as "PowerConsumption",
"Diameter" or similar.</xs:documentation>
            </xs:annotation>
        </xs:attribute>
        <xs:anyAttribute processContents="lax">
            <xs:annotation>
                <xs:documentation>Proprietary or extended
attributes.</xs:documentation>
            </xs:annotation>
        </xs:anyAttribute>
    </xs:complexType>
    <xs:complexType name="MetaDataType">
        <xs:sequence>
            <xs:element name="InfoItem" type="InfoItemType" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="DescriptionType">
        <xs:annotation>
            <xs:documentation>String with some"human-readable"
text.</xs:documentation>
        </xs:annotation>
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute name="lang" type="xs:string" use="optional">
                    <xs:annotation>
                        <xs:documentation>Language of "description"
text.</xs:documentation>
                    </xs:annotation>
                </xs:attribute>
                <xs:anyAttribute processContents="lax">
                    <xs:annotation>
                        <xs:documentation>Proprietary or extended
attributes.</xs:documentation>
                    </xs:annotation>
                </xs:anyAttribute>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
    <xs:complexType name="QlmIDType">
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute name="idType" type="xs:string" use="optional">
                    <xs:annotation>
                        <xs:documentation>Text identifying the ID
schema.</xs:documentation>
                    </xs:annotation>
                </xs:attribute>
                <xs:attribute name="tagType" type="xs:string" use="optional">
                    <xs:annotation>
                        <xs:documentation>Text identifying the ID Tag media
type. </xs:documentation>
                    </xs:annotation>
                </xs:attribute>
```

```
                <xs:attribute name="startDate" type="xs:dateTime"
use="optional">
                    <xs:annotation>
                        <xs:documentation>Start of validity for the
ID</xs:documentation>
                    </xs:annotation>
                </xs:attribute>
                <xs:attribute name="endDate" type="xs:dateTime" use="optional">
                    <xs:annotation>
                        <xs:documentation>End of validity for the
ID</xs:documentation>
                    </xs:annotation>
                </xs:attribute>
                <xs:anyAttribute processContents="lax">
                    <xs:annotation>
                        <xs:documentation>Proprietary
attributes.</xs:documentation>
                    </xs:annotation>
                </xs:anyAttribute>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
    <xs:complexType name="ValueType">
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute name="type" use="optional" default="xs:string">
                    <xs:simpleType>
                        <xs:restriction base="xs:string"/>
                    </xs:simpleType>
                </xs:attribute>
                <xs:attribute name="dateTime" type="xs:dateTime"
use="optional"/>
                <xs:attribute name="unixTime" type="xs:long" use="optional"/>
                <xs:anyAttribute processContents="lax"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:schema>
```

B Example O-DF Structures

This appendix shows example messages for some basic cases. More examples are normally available at the website(s) where this specification is published.

Example 5: Object->Object->InfoItem->Value(s)

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Example of Object->Object->InfoItem->value(s).  -->
<Objects  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="odf.xsd">
    <Object type="someType">
        <id>UniqueTargetID_1</id>
        <InfoItem name="InfoItem1">
            <value>Value1</value>
            <value>Value2</value>
            <value>Value3</value>
        </InfoItem>
        <InfoItem name="InfoItem2">
            <value>Value</value>
        </InfoItem>
        <Object type="someType">
            <id>SubTarget1</id>
            <InfoItem name="SubInfoItem1">
            </InfoItem>
            <Object type="someType">
                <id>SubSubTarget1</id>
                <InfoItem name="SubSubTarget1InfoItem1">
                    <value>22.5</value>
                </InfoItem>
            </Object>
        </Object>
        <Object type="someType">
            <id>SubTarget2</id>
            <InfoItem name="SubTarget2InfoItem1">
                <value>34.6</value>
            </InfoItem>
        </Object>
    </Object>
</Objects>
```

Example 6: Metadata about Refrigerator Power Consumption InfoItem

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Example of a simple "odf" structure for a refrigerator. -->
<Objects xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="odf.xsd">
    <Object>
        <id>SmartFridge22334411</id>
        <InfoItem name="PowerConsumption">
            <MetaData>
                <InfoItem name="format"><value
type="xs:string">xs:double</value></InfoItem>
                <InfoItem name="latency"><value
type="xs:int">5</value></InfoItem>
```

```
                <InfoItem name="readable"><value
type="xs:boolean">true</value></InfoItem>
                <InfoItem name="writable"><value
type="xs:boolean">false</value></InfoItem>
                <InfoItem name="unit"><value
type="xs:string">Watts</value></InfoItem>
                <InfoItem name="accuracy"><value
type="xs:double">1</value></InfoItem>
            </MetaData>
        </InfoItem>
    </Object>
</Objects>
```

Example 7: Measurement Values for Refrigerator Power Consumption

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Example of a simple "odf" structure for a refrigerator. -->
<Objects xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="odf.xsd">
    <Object type="Refrigerator Assembly Product">
        <id>SmartFridge22334411</id>
        <InfoItem name="Consumed Electrical Power Measure">
            <description>Power consumption values with timestamp.</description>
            <value dateTime="2001-10-26T15:33:21">15.5</value>
            <value dateTime="2001-10-26T15:33:50">15.7</value>
            <value dateTime="2001-10-26T15:34:15">1.3</value>
            <value dateTime="2001-10-26T15:34:35">1.5</value>
            <value dateTime="2001-10-26T15:34:52">15.3</value>
        </InfoItem>
    </Object>
</Objects>
```

C JSON Mapping and Examples

Many XML generation and manipulation tools can convert XML messages into JavaScript Object Notation (JSON), and *vice versa*. There are also many frameworks that can do the needed conversions in “real-time”. This example shows Example 7 converted into JSON.

```
{
    "XML": {
        "version": 1.0,
        "encoding": "UTF-8"
    },
    "Comment": " Example of a simple “odf” structure for a refrigerator. ",
    "Objects": {
        "xmlns:xsi": "http://www.w3.org/2001/XMLSchema-instance",
        "xsi:noNamespaceSchemaLocation": "odf.xsd",
        "Object": {
            "type": "Refrigerator Assembly Product",
            "id": "SmartFridge22334411",
            "InfoItem": {
                "name": "Consumed Electrical Power Measure",
                "description": "Power consumption values with timestamp.",
                "value": [
                    {
                        "dateTime": "2001-10-26T15:33:21",
                        "Text": 15.5
                    }, {
                        "dateTime": "2001-10-26T15:33:50",
                        "Text": 15.7
                    }, {
                        "dateTime": "2001-10-26T15:34:15",
                        "Text": 1.3
                    }, {
                        "dateTime": "2001-10-26T15:34:35",
                        "Text": 1.5
                    }, {
                        "dateTime": "2001-10-26T15:34:52",
                        "Text": 15.3
                    }
                ]
            }
        }
    }
}
```

Abbreviations

HTTP	Hypertext Transfer Protocol
IoT	Internet of Things
JMS	Java Message Service
JSON	JavaScript Object Notation
O-DF	Open Data Format
PMI	PROMISE Messaging Interface
QLM	Quantum Lifecycle Management
REST	REpresentational State Transfer
RFID	Radio-Frequency Identification
SOAP	Simple Object Access Protocol
SSCC	Serial Shipping Container Code
URL	Uniform Resource Locator
WSDL	Web Services Description Language
XML	EXtensible Markup Language