

## Identity Attribute Metadata - Draft 01

### Abstract

An outline of the attribute metadata schema and types for identity attributes.

---

### Table of Contents

- 1. Overview
- 2. Terminology
  - 2.1. Definitions and Conventions
- 3. Metadata Format
  - 3.1. Data Format Types
  - 3.2. Attribute Types
    - 3.2.1. Standard Predicates
    - 3.2.2. Supplemental Predicates
    - 3.2.3. Example
- 4. Future Directions
  - 4.1. Compound Properties
  - 4.2. Equivalents
  - 4.3. Higgins Ontology Predicates
- 5. References
  - 5.1. Normative References
  - 5.2. Informative References
- § Author's Address

---

### 1. Overview

This document defines the schema used to describe identity object data as used in such protocols as OpenID Attribute Exchange. The schema data is intended to be resolvable at the URI of the identity object, the attribute type identifier.

---

## 2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] (Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," March 1997.).

---

### 2.1. Definitions and Conventions

#### Identity Object

Identity "objects" (IdO) are defined to be sets of name-value pairs of personal identity information. They assert the claim that, for a given identity subject, a named property has the provided value.

#### Service Provider

Service providers (SPs) are entities that require identification information.

#### Identity Provider

An identification provider (IdP) stores IdOs, which are made available to an identity agent upon request.

#### Identity Agent

The identity agent (IdA) operates under the user's control and intermediates requests for ID objects between the identification provider and the SP.

#### Identity Attribute Type

Identity attribute types (also referred to as simply "attribute types") are types of subject properties expressed in an identity context. Examples are "surname" or "birth date".

#### Identity Attribute Format Type

The identity attribute format type ("format type") refers to the layout of the data in the value of an identity attribute type. They may be as simple as a normalized string or as complicated as a telephone number format.

---

## 3. Metadata Format

---

### 3.1. Data Format Types

Data format types are the primitive types used to define the layout of the information in the attribute values. The types are defined in XML Schema ([W3C.REC-xmlschema-2-20041028] (Biron, P. and A. Malhotra, "XML Schema Part 2: Datatypes Second Edition," October 2004.)) and include the predefined XML Schema types.

For example, here are several formats used with the OpenID attribute exchange protocol ([OpenID.attribute-exchange-1.0] (Hardt, D., "OpenID Attribute Exchange," November 2006.)).

<http://schema.openid.net/types/country>

```
<xsd:simpleType name="country">
```

2

```

<xsd:restriction base="xsd:NMTOKEN">
  <!-- AFGHANISTAN -->
  <xsd:enumeration value="AF"/>
  <!-- ÅLAND ISLANDS -->
  <xsd:enumeration value="AX"/>
  <!-- ALBANIA -->
  <xsd:enumeration value="AL"/>
  <!-- ALGERIA -->
  <xsd:enumeration value="DZ"/>
  ... etc ...
</xsd:restriction>
</xsd:simpleType>

```

<http://schema.openid.net/types/email>

```

<xsd:simpleType name="email" >
  <xsd:restriction base="xsd:token">
    <xsd:pattern value=
      "([\.a-zA-Z0-9_-])+@
      ([a-zA-Z0-9_-])+((\.[a-zA-Z0-9_-])+)+"
    />
  </xsd:restriction>
</xsd:simpleType>

```

<http://schema.openid.net/types/gender>

```

<xsd:simpleType name="gender">
  <xsd:restriction base="xsd:NMTOKEN">
    <xsd:enumeration value="M"/>
    <xsd:enumeration value="F"/>
  </xsd:restriction>
</xsd:simpleType>

```

<http://schema.openid.net/types/phone>

```

<xsd:simpleType name="phone">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="\+?[0-9]+(\.[0-9]+)*"/>
  </xsd:restriction>
</xsd:simpleType>

```

<http://schema.openid.net/types/timezone>

```

<xsd:simpleType name="timezone">
  <xsd:restriction base="xsd:string">
    <xsd:pattern value="[a-zA-Z/-]+" />
  </xsd:restriction>
</xsd:simpleType>

```

---

## 3.2. Attribute Types

The metadata for attribute types is expressed in RDF/XML format. It is intended to be a fairly simple way of expressing the minimal amount of metadata needed to describe the attribute types. More complicated ontological data is optional but recommended.

Each attribute type record should include a stylesheet XML directive pointing to an XSL template that translates the metadata into a human readable format.

---

### 3.2.1. Standard Predicates

The standard predicates that MUST be in all metadata records are:

<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>

The `rdf:type` predicate has as its object the XML Schema data type or a type defined as per Section 3.1 (Data Format Types).

<http://www.w3.org/2000/01/rdf-schema#label>

The label is a short description of the attribute type. XML provides an `xml:lang` attribute that can be used on this element to provide a way to describe the language as per [RFC4646] (Phillips, A. and M. Davis, "Tags for Identifying Languages," September 2006.) used for the content of the element. Using language tagging in this way, multiple labels can be provided for localization purposes.

<http://www.w3.org/2000/01/rdf-schema#comment>

The `rdfs:comment` element is used to provide a long textual description of the attribute type. As for the `rdf:label` element, multilingual documentation is supported by the language tagging feature of RDF literals.

---

### 3.2.2. Supplemental Predicates

These predicates are optional and MAY be included in metadata records:

<http://schema.openid.net/metadata#example>

Example value data for the attribute type.

<http://www.w3.org/2000/01/rdf-schema#seeAlso>

Indicates a resource that might provide additional information about the subject attribute type.

<http://schema.openid.net/metadata#acquisition>

The object of this predicate is a URL from which the IdO may be acquired. Multiple URLs may be specified. The acquisition mechanism is not specified, but would be retrieved using a discovery mechanism specific to the protocol being used.

<http://schema.openid.net/metadata#authority>

Except in the case of a self-asserted IdO, a list of authority URIs for asserted claims is necessary. Each URI is that of an assertion authority that is allowed to make the IdO claim.

### 3.2.3. Example

A brief example of the standard predicates and the openid:example element as applied to the

`http://schema.openid.net/namePerson/first` attribute type.

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="schema.xslt"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:openid="http://schema.openid.net/metadata#">
<rdf:Description rdf:about="http://schema.openid.net/namePerson/first">
  <rdfs:label>
    First name
  </rdfs:label>
  <rdfs:comment>
    First or given name of subject
  </rdfs:comment>
  <openid:example>
    John
  </openid:example>
  <rdf:type
    rdf:resource="http://www.w3.org/2001/XMLSchema#normalizedString"/>
  <openid:acquisition
    rdf:resource="http://example.gov/id"/>
</rdf:Description>
</rdf:RDF>
```

---

## 4. Future Directions

Additional metadata information may be added as more complex attribute types are constructed. The following sections outline possible extensions to the existing simple type definitions.

---

### 4.1. Compound Properties

The IdO may also be composed of an aggregate of other IdO types, in which case the aggregate IdO URIs will be referenced.

---

### 4.2. Equivalentents

An IdO may make a claim that is equivalent to the claim of an IdO of a different type. The equivalent IdO types are listed in this section.

An IdO may be transformed to one of a different type if it is listed as an equivalent. This property is not commutative.

This information may be extended to include translation mechanisms between format types. A richer transform specification would allow claims to be made based on a broader equivalence domain.

---

### 4.3. Higgins Ontology Predicates

The Higgins project has created a base ontological vocabulary at [Higgins-Ontology] (Trevithick, P., "Higgins Ontology v1.10," October 2006.). Use of this vocabulary allows for the integration of the attribute types into a broader catalog.

---

## 5. References

---

### 5.1. Normative References

- |                                 |   |
|---------------------------------|---|
| [OpenID.attribute-exchange-1.0] | Hardt, D., "OpenID Attribute Exchange," November 2006 (TXT, HTML).  |
| [RFC2119]                       | Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," BCP 14, RFC 2119, March 1997 (TXT, HTML, XML).   |
| [RFC4646]                       | Phillips, A. and M. Davis, "Tags for Identifying Languages," BCP 47, RFC 4646, September 2006.  |
| [W3C.REC-xmlschema-2-20041028]  | Biron, P. and A. Malhotra, "XML Schema Part 2: Datatypes Second Edition," World Wide Web Consortium Recommendation REC-xmlschema-2-20041028, October 2004 (HTML). |

---

### 5.2. Informative References

- |                    |   |
|--------------------|---|
| [Higgins-Ontology] | Trevithick, P., "Higgins Ontology v1.10," October 2006. |
|--------------------|---|
- 

### Author's Address

Dick Hardt  
Sxip Identity  
798 Beatty Street  
Vancouver, BC V6B 2M1  
CA  
Email: [dick@sxip.com](mailto:dick@sxip.com)  
URI: <http://sxip.com/>