

# Attribute Registry Overview

Version 1.1 of the Attribute Registry is the latest release of one of the early deliverables from the Scalable Privacy project.

The core data elements are attributes, each of which comes from one of a defined set of specifications or standards. The images in this overview were taken directly from the ontology tool, Protégé, used to maintain the registry. A web-accessible version of the registry is available at <http://webprotege.stanford.edu/#Edit:projectId=623e0196-844a-4d86-a65b-b4d2f923ab97>

- Specifications
- Attributes
- Attribute Class
- Future Work
- Reference: The Currently Defined Set of Sixty-one Attribute Classes

## Specifications

The screenshot shows the Protégé ontology editor interface. On the left, the 'Class hierarchy: Specification' pane shows a tree structure with 'Thing' as the root, followed by 'Attribute', 'AttributeClass', and 'Specification' (highlighted in blue). On the right, the 'Annotations: Specification' pane is empty. Below it, the 'Description: Specification' pane shows a list of members. Each member is preceded by a diamond icon and followed by three circular icons: a question mark, an at-sign, and an X.

Member	Question Mark	@	X
eduCourse	?	@	X
eduMember	?	@	X
eduPerson	?	@	X
FICAM	?	@	X
IMS_Person	?	@	X
OpenID_Connect	?	@	X
OpenSocial	?	@	X
RFC4524	?	@	X
Schac	?	@	X
SCIM_Core	?	@	X
VA_EDM	?	@	X
X.520	?	@	X

Each attribute of the 715 attributes currently in the registry is associated with the specification that defines it. For example, the OpenID Connect specification covers the following attributes:

Class hierarchy: Sp

- Thing
  - Attribute
  - AttributeClass
  - Specification

Annotations: OpenID\_Connect

Annotations +

Description: OpenID\_Conn

Types +

- Specification ? @ x o

Same Individual As +

Different Individuals +

Property assertions: OpenID\_Connect

Object property assertions +

- specifies oidc-website ? @ x o
- specifies oidc-preferred\_username ? @ x o
- specifies oidc-region ? @ x o
- specifies oidc-phone\_number ? @ x o
- specifies oidc-formatted ? @ x o
- specifies oidc-middle\_name ? @ x o
- specifies oidc-profile ? @ x o
- specifies oidc-nickname ? @ x o
- specifies oidc-gender ? @ x o
- specifies oidc-locality ? @ x o
- specifies oidc-family\_name ? @ x o
- specifies oidc-zoneinfo ? @ x o
- specifies oidc-country ? @ x o
- specifies oidc-locale ? @ x o
- specifies oidc-address ? @ x o
- specifies oidc-name ? @ x o
- specifies oidc-picture ? @ x o
- specifies oidc-email\_verified ? @ x o
- specifies oidc-user\_id ? @ x o
- specifies oidc-street\_address ? @ x o
- specifies oidc-email ? @ x o
- specifies oidc-given\_name ? @ x o
- specifies oidc-updated\_time ? @ x o
- specifies oidc-postal\_code ? @ x o
- specifies oidc-birthday ? @ x o

Object property hierarchy

Object property hierarchy

- topObjectProperty
  - classifies
  - isClassifiedBy
  - isSpecifiedBy
  - specifies

## Attributes

An example attribute entry in the registry appears as follows (this is the Profile attribute from OpenID Connect):

Description: oidc-profile

Types +

- Attribute ? @ x o

Same Individual As +

Different Individuals +

Property assertions: oidc-profile

Object property assertions +

- isSpecifiedBy OpenID\_Connect ? @ x o
- isClassifiedBy Profile ? @ x o

Data property assertions +

- AttributeName "profile"^^string ? @ x o
- SourceLocation "http://openid.net/specs/openid-connect-basic-1\_0.html#id\_res"^^string ? @ x o
- Syntax "string"^^string ? @ x o
- Section "2.52"^^string ? @ x o
- Definition "URL of the End-User's profile page."^^string ? @ x o

Note the metadata (Object properties and Data properties) recorded in the registry for the Profile attribute. Version 1.0 contains a minimal set of metadata elements. Other types of metadata may be added to suit emerging needs in the attribute ecosystem work.

Another example attribute entry in the registry is eduPersonPrincipalName from the eduPerson specification:

Property assertions: eduPersonPrincipalName

Object property assertions +

- isClassifiedBy Identifier
- isSpecifiedBy eduPerson

Data property assertions +

- SourceLocation "http://macedir.org/specs/eduperson/#eduPersonPrincipalName"^^string
- AttributeName "eduPersonPrincipalName"^^string
- Oid "1.3.6.1.4.1.5923.1.1.1.6"^^string
- Definition "scoped identifier for a person. It should be represented in the form 'user@scope' where 'user' is a name-based identifier for the person and where 'scope' defines a local security domain. Each value of 'scope' defines a namespace within which the assigned identifiers MUST be unique. Given this rule, if two eduPersonPrincipalName (ePPN) values are the same at a given point in time, they refer to the same person"^^string
- Syntax "directoryString"^^string
- Section "2.2.8"^^string

Note the relatively full Definition element in this case. This is drawn from the specification itself.

### Attribute Class

At the top of the eduPersonPrincipalName example above, there is the object property "isClassifiedBy" with the value "Identifier". This is an example of a metadata element meant to categorize attributes across specifications into a defined set of types. This metadata element is called "Attribute Class". Here is the first part of a listing of the currently defined attribute classes:

Class hierarchy: Attr

- Thing
  - Attribute
  - AttributeClass
  - Specification

Annotations: AttributeClass

Annotations +

Description: AttributeClass

Members +

- Account
- Address
- Affiliation
- Application
- Assurance
- Birthdate
- BirthPlace
- Citizenship
- Contact
- Country
- Date

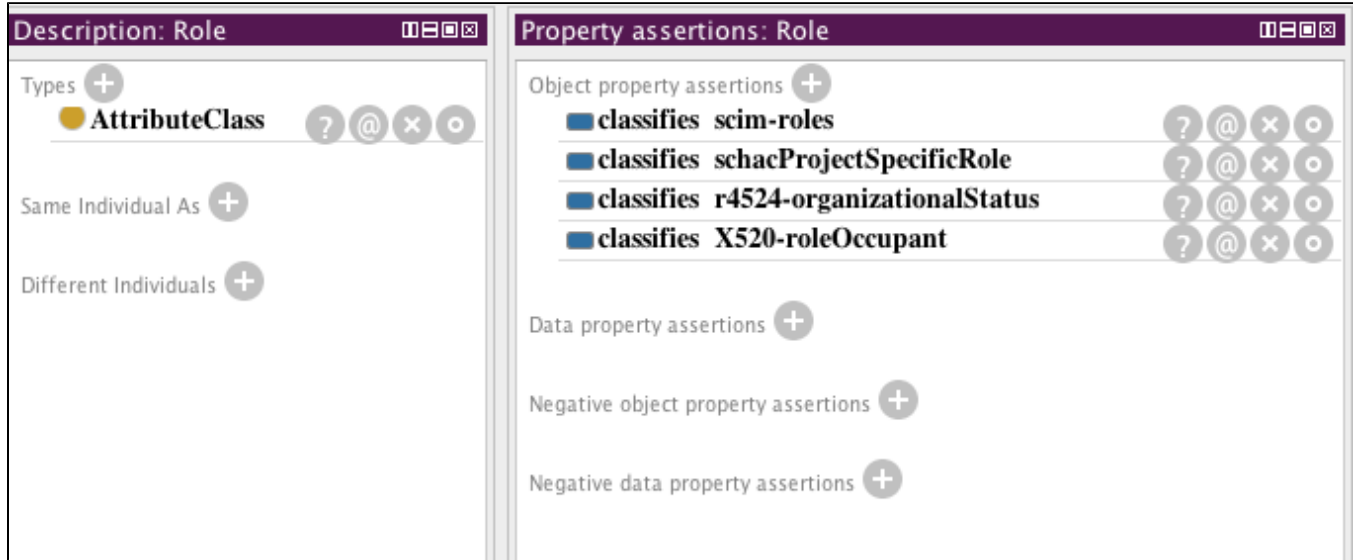
A couple examples will clarify the notion of attribute class. Take the example of attributes relating to preferences.

The screenshot shows two panels. The left panel, titled 'Description: Preference', contains a 'Types' section with a yellow circle next to 'AttributeClass' and icons for help, search, close, and refresh. Below are sections for 'Same Individual As' and 'Different Individuals', each with a plus sign. The right panel, titled 'Property assertions: Preference', has sections for 'Object property assertions', 'Data property assertions', 'Negative object property assertions', and 'Negative data property assertions'. Under 'Object property assertions', there are three entries: 'classifies osoc-emails-primary', 'classifies osoc-contactPreference', and 'classifies r4524-drink'. Each entry has icons for help, search, close, and refresh.

The Open Social specification (attributes whose prefix is "osoc") contains two attributes in the class "preference". The LDAP specification (currently RFC4524) contains an attribute "drink" which indicates personal preference as well. Here is the preference attribute, osoc-emails-primary:

The screenshot shows two panels. The left panel, titled 'Description: osoc-emails-', contains a 'Types' section with a yellow circle next to 'Attribute' and icons for help, search, close, and refresh. Below are sections for 'Same Individual As' and 'Different Individuals', each with a plus sign. The right panel, titled 'Property assertions: osoc-emails-primary', has sections for 'Object property assertions' and 'Data property assertions'. Under 'Object property assertions', there are two entries: 'isClassifiedBy Preference' and 'isSpecifiedBy OpenSocial'. Under 'Data property assertions', there are four entries: 'AttributeName "emails-primary"^^string', 'Definition "A Boolean value indicating whether this instance of the Plural Field is the primary or preferred value of for this field, e.g. the preferred mailing address or primary e-mail address. Service Providers MUST NOT mark more than one instance of the same Plural Field as primary='true', and MAY choose not to mark any fields as primary, if this information is not available. For efficiency, Service Providers SHOULD NOT mark all non-primary fields with primary='false', but should instead omit this sub-field for all non-primary instances."^^string', 'Syntax "Boolean"^^string', 'SourceLocation "http://opensocial-resources.googlecode.com/svn/spec/trunk/Social-Data.xml#Person"^^string', and 'Section "2.14"^^string'. Each entry has icons for help, search, close, and refresh.

Another example of the attribute class metadata is "role". Several specifications (SCIM, SCHAC, LDAP and X.520) contain attributes meant to carry some definition of a person's role:



### Future Work

Going forward, attributes from additional specifications and standards will be added (including schema from additional national federations in Norway and Australia). One open issue is whether the current list of attribute metadata is adequate or whether there would be value in carrying additional metadata elements in a general purpose registry of this sort.

### Reference: The Currently Defined Set of Sixty-one Attribute Classes

AccessibilityProfile, Account, Address, Affiliation, Age, Anniversary, Application, Assurance  
 Birthdate, BirthPlace  
 Certification, Citizenship, Clearance, Contact, Country, Course  
 DN, Date, DeathDate, Description  
 Email, EmailMetadata, Entitlement, Ethnicity  
 Format  
 Gender, Group  
 HostName  
 Identifier  
 Jurisdiction  
 Language, Link, Locale, Location  
 Manager  
 Name, Note  
 Operational, Organization  
 Password, Phone, Photo, PhysicalCharacteristic, Pointer, Position, Preference, Presence, Privacy, Profile  
 Relationship, Residency, Role  
 Salutation, Search, StateOrProvince, StatusMessage, SuperiorNode  
 Tag, Timezone  
 URL  
 VitalEvent