

ComputedIdConnector

The `ComputedId` data connector generates an attribute from the (usually SHA-1) digest of the requesting entityID, an attribute value, and a salt that must be kept secret to prevent off-line generation of the hashes to recover the underlying attribute value.

The attribute value is therefore opaque and unique per user, per relying party, suitable for use as a SAML "persistent" NameID or "pairwise-id" Subject Attribute.

Schema Name and Location

This `xsi:type` is defined by the `urn:mace:shibboleth:2.0:resolver` schema^{3.3}, located at <http://shibboleth.net/schema/idp/shibboleth-attribute-resolver.xsd>

Prior to V3.3 supplied plugins were defined by a schema type (`xsi:type`) in the `urn:mace:shibboleth:2.0:resolver:dc` namespace, the schema for which is located at <http://shibboleth.net/schema/idp/shibboleth-attribute-resolver-dc.xsd>. This is still supported, but every element or type in the `urn:mace:shibboleth:2.0:resolver:dc` namespace has an equivalently named (but not necessarily identical) version in the `urn:mace:shibboleth:2.0:resolver` namespace. The use of the `urn:mace:shibboleth:2.0:resolver` namespace also allows a relaxation of the ordering requirements of child elements to reduce strictness.\

Reference

Attributes

Any of the [common attributes](#) can be specified. In addition the following attributes are supported:

Name	Type	Default	Description
<code>generatedAttributeID</code>	string	ID of the connector	The id of the <code>IdPAttribute</code> that is generated
<code>sourceAttributeID</code>	string		<i>DEPRECATED in V3.4</i> The id of the <code>IdPAttribute</code> used as input to the computed ID, required for older versions
<code>salt</code>	string, required		A salt, of at least 16 bytes, used in the computed ID
<code>encoding</code> ^{3.3.2}	string	BASE64	Controls the eventual text encoding of the value, this should be set to "BASE32" for new deployments (see the warning box about case sensitivity under PersistentNameIDGenerationConfiguration)
<code>algorithm</code> ^{3.4}	string	SHA	Controls the digest algorithm applied



Configuring salt prior to V3.3

Prior to release 3.3 the parser mishandled the provided salt and stripped trailing and leading spaces from it, see case [IDP-982](#). This rendered the values incompatible with those used in V2.

A workaround is to indirect through a property: for instance:

Attribute-resolver.xml:

```
<DataConnector id="computed" xsi:type="ComputedId"
  sourceAttributeID="theSourceRemainsTheSame"
  generatedAttributeID="Foo"
  salt="%{idp.persistentId.salt}">
```

idp.properties

```
idp.persistentId.salt = String with Spaces before and after
```

Child Elements

Any of the [common child elements](#) can be specified.

Examples

The example produces a hashed value using an input value "Foo" from a DataConnector named "DataSourceForFoo".

```
<DataConnector id="ComputedIDConnector" xsi:type="ComputedId"
  generatedAttributeID="ComputedID"
  salt="abcdefghijklmnopqrstuvwxy"
  encoding="BASE32">

  <InputDataConnector ref="DataSourceForFoo" attributeNames="Foo" />

</DataConnector>
```