attribute-resolver.xml ファイルの変更（IdPv4）

attribute-resolver.xml ファイルの変更

0. 事前準備

デフォルトの /opt/shibboleth-idp/conf/attribute-resolver.xml をテンプレートで差し替え、下記にしたがい修正してください。

1．属性の定義部分を、以下の例に従い有効にします

各属性の定義方法はここを参照下さい。

(1) eduPersonPrincipalNameを有効とする例：

```
<!-- Definition for eduPersonPrincipalName -->
<!-- --> ← コメント終了を追加して、以下を有効とします
<AttributeDefinition xsi:type="Scoped" id="eduPersonPrincipalName" scope="%{idp.scope}">
          ↑ scopeはidp.propertiesを参照するので設定不要です。
          ↓ attributeNamesにLDAP内の属性名を設定します。
<InputDataConnector ref="myLDAP" attributeNames="uid" />
</AttributeDefinition>
<!-- --> ← コメント開始を追加して、上記を有効とします
```

(2) eduPersonEntitlementのurn:mace:dir:entitlement:common-lib-termsを有効とする例：

```
<!-- Definition for eduPersonEntitlement -->
<!-- --> ← コメント終了を追加して、以下を有効とします
<AttributeDefinition xsi:type="Simple" id="eduPersonEntitlement">
<InputDataConnector ref="staticEntitlementCommonLibTerms" attributeNames="eduPersonEntitlement" />
</AttributeDefinition>
<!-- --> ← コメント開始を追加して、上記を有効とします
<!-- Static Connector for eduPersonEntitlement (common-lib-terms) -->
<!-- --> ← コメント終了を追加して、以下を有効とします
<DataConnector id="staticEntitlementCommonLibTerms" xsi:type="Static">
<Attribute id="eduPersonEntitlement">
<Value>urn:mace:dir:entitlement:common-lib-terms</Value>
</Attribute>
</DataConnector>
<!-- --> ← コメント開始を追加して、上記を有効とします
```

※属性値としてcommon-lib-terms以外を送信する場合は、しかるべき人にそのような値を生成するDataConnectorないしAttributeDefinitionを用意し、上記4行目と5行目の間にInputDataConnectorないしInputAttributeDefinitionとして追加してください。

2．利用するコネクタを設定します

現在学認で配布しているバージョンでは ldap.properties を参照するようになっていますので、LDAP URLやbaseDN等を直接記述する必要はありません。ただし returnAttributes を指定している場合もしくは useStartTLS=true もしくは useSSL=true としている場合はテンプレートファイル中のコメントに従って修正してください。また後者の場合は ldap.properties の idp.authn.LDAP.sslConfig が certificateTrust もしくはコメントアウトされていることを確認してください。他の設定では意図した動作をしません。

また、LDAPに登録されている属性をそのまま送出する場合は個別にSimple AttributeDefinitionを定義するのではなく、LDAP DataConnectorの exportAttributes にその属性名を列挙する形になっています。必要であれば<Column>による属性名の変換を併用してください。詳細はテンプレートファイル中のコメントを参照してください。万が一LDAPからダイレクトに送出すべき属性がない場合は、exportAttributes="" の部分を削除してください。

LDAPコネクタが有効化されていない場合は有効化してください。

```
<!-- ======================================== -->
<!-- Data Connectors -->
<!-- ======================================== -->

<!-- Example LDAP Connector -->
<!--
Add the attributes that you want to send directly to the exportAttributes
attribute. the exportAttributes is a space-separated list, which can contain:
mail sn ou givenName displayName eduPersonAssurance eduPersonOrcid
jasn jaGivenName jaDisplayName jaou

e.g.
exportAttributes="mail sn ou givenName displayName jasn jaGivenName jaDisplayName jaou"

If you have an LDAP attribute name which is different from the above list,
change the columnName attribute in the corresponding Column element
approprietely and enable that line.

e.g.
<Column columnName="email" attributeID="mail" />


Add the following configuration below the <DataConnector line in order
to use StartTLS (i.e. idp.attribute.resolver.LDAP.useStartTLS in
conf/ldap.properties is true) or LDAPS (i.e. you use "ldaps:" protocol
in idp.attribute.resolver.LDAP.ldapURL in conf/ldap.properties):

trustFile="%{idp.attribute.resolver.LDAP.trustCertificates}"
-->
<!-- -->
<DataConnector id="myLDAP" xsi:type="LDAPDirectory"
ldapURL="%{idp.attribute.resolver.LDAP.ldapURL}"
baseDN="%{idp.attribute.resolver.LDAP.baseDN}"
principal="%{idp.attribute.resolver.LDAP.bindDN}"
principalCredential="%{idp.attribute.resolver.LDAP.bindDNCredential}"
useStartTLS="%{idp.attribute.resolver.LDAP.useStartTLS:true}"
connectTimeout="%{idp.attribute.resolver.LDAP.connectTimeout}"
responseTimeout="%{idp.attribute.resolver.LDAP.responseTimeout}"
exportAttributes="CHANGETHISTOATTRIBUTESYOUWANTTOSEND"
noResultIsError="%{idp.attribute.resolver.LDAP.noResultIsError:true}">
<FilterTemplate>
<![CDATA[
%{idp.attribute.resolver.LDAP.searchFilter}
]]>
</FilterTemplate>
<ConnectionPool
minPoolSize="%{idp.pool.LDAP.minSize:3}"
maxPoolSize="%{idp.pool.LDAP.maxSize:10}"
blockWaitTime="%{idp.pool.LDAP.blockWaitTime:PT3S}"
validatePeriodically="%{idp.pool.LDAP.validatePeriodically:true}"
validateTimerPeriod="%{idp.pool.LDAP.validatePeriod:PT5M}"
validateDN="%{idp.pool.LDAP.validateDN:}"
validateFilter="%{idp.pool.LDAP.validateFilter:(objectClass=*)}"
expirationTime="%{idp.pool.LDAP.idleTime:PT10M}"/>
<!--
<Column columnName="mail" attributeID="mail" />
<Column columnName="sn" attributeID="sn" />
<Column columnName="ou" attributeID="ou" />
<Column columnName="givenName" attributeID="givenName" />
<Column columnName="displayName" attributeID="displayName" />
<Column columnName="eduPersonAssurance" attributeID="eduPersonAssurance" />
<Column columnName="eduPersonOrcid" attributeID="eduPersonOrcid" />
<Column columnName="sn;lang-ja" attributeID="jasn" />
<Column columnName="givenName;lang-ja" attributeID="jaGivenName" />
<Column columnName="displayName;lang-ja" attributeID="jaDisplayName" />
<Column columnName="ou;lang-ja" attributeID="jaou" />
-->
</DataConnector>
<!-- -->
```

※ attribute-resolver.xmlに記述する内容に '&' や '<' を含む場合は '&' や '<' のように記述してください。逆にldap.propertiesに記述する場合はそのまま記述してください。

→ meatwiki:GakuNinShare:トラブルシューティング

ComputedIdやStoredIdのコネクタも同様ですが、sourceAttributeやsaltは saml-nameid.properties ファイルや `credentials/secrets.properties` を参照するようになっていますので、そちらで設定してください。
その他詳細はテンプレートファイル中のコメントをご参照ください。

３．学認で規定している属性を追加します

gakunin-rules.tar.gzをダウンロードして、/opt/shibboleth-idp/conf/配下に展開します。（/opt/shibboleth-idp/conf/attributes/custom/以下に*.propertiesを配置）

```
# tar zxv -C /opt/shibboleth-idp/conf -f gakunin-rules.tar.gz
```

BACK TOP NEXT