LDAPPC 1.5.0 Example Configuration

Unable to render {include} The included page could not be found.

LDAPPC 1.5.0 Example Configuration

As of (pending) version 1.5.0, LDAPPC :

- incorporates the Shibboleth Attribute Resolver for the calculation of attributes to be provisioned

- functions as an SPML 2 Provisioning Service Provider

- provisions LDAP and other targets for which an SPML 2 provider interface can be written

- uses vt-ldap 3.0 for LDAP communication

All attributes to be provisioned are calculated by the Shibboleth Attribute Resolver. Grouper specific DataConnectors are provided which return Group, Member, and Stem data. Additional DataConnectors may be included in the Attribute Resolver configuration to include sources other than Grouper. A syntax for accessing Grouper data as attributes from within the Attribute Resolver configuration has been defined.

Somewhat in violation of the SPML specification, targets to be provisioned are assumed to also be *providers*. A Provisioning Service Provider is defined as "a software component that listens for, processes, and returns the results for well-formed SPML requests". Although the SPML specification states (in bold) that "a target is not a provider", LDAPPC assumes that targets are able to process SPML requests. In other words, LDAPPC uses SPML as the common language for communicating with targets.

Given a string which identifies a Group, Member, or Stem, LDAPPC requests attributes from a Shibboleth Attribute Authority and Attribute Resolver. A simple Attribute Authority is provided which allows for filtering. These attributes are used to create a representation of the Provisioning Service Object to be provisioned. For our purposes, an object consists of an identifier, attributes, and references. A reference is essentially an attribute whose value is the identifier of another object.

The configuration of LDAPPC has changed significantly, is now modeled after Shibboleth, and will likely consist of :

ldappc.xml

ldappc-resolver.xml

ldappc-services.xml

ldappc-internal.xml

ldappc-ldap.xml

All attributes to be provisioned for all targets and objects are defined in a single Attribute Resolver configuration to reduce the number of queries made to Grouper.

Configuration Example : ldappc.xml

The ldappc.xml configuration file defines the targets and objects to be provisioned.

A single target example follows :

```
<target id="openldap" provider="provider-openldap" />

<object id="group">
  <identifier id="group-dn" />
  <attribute name="objectclass" />
  <attribute name="cn" />
  <attribute name="hasMember" id="hasMember" />
  <reference name="member" id="members-jdbc" toPSOId="member" />
  <reference name="member" id="members-g:gsa" toPSOId="group" />
</object>

<object id="member">
   <identifier id="member-dn" />
   <attribute name="isMemberOf" id="member-isMemberOf" />
</object>
```

<target>

The target is "openldap", which contains two objects (SPML schema entities), "group" and "member".

<object>

Each object consists of an identifier, and optionally attributes and references. The value or values of each identifier, attribute, and reference are defined by the Shibboleth Attribute Definition with the corresponding id. The name of the provisioned attribute or reference is defined by the "name" attribute. If not specified, the "id" defaults to the "name" attribute.

The "openldap" target's provider service is "provider-openldap" as defined in */dappc-services.xml/* :

```
<Service id="provider-openldap" xsi:type="ldappc:LdapPoolProvider" ldapPoolId="ldapPool">
  <ConfigurationResource file="ldap.xml" xsi:type="resource:ClasspathResource" />
</Service>
```

And, this provider is based on a vt-ldap 3.0 pool as defined in */dappc-ldap.xml/* :

```
...
<bean id="ldapPool"
    class="edu.vt.middleware.ldap.pool.SoftLimitLdapPool"
    init-method="initialize"
...
```

See http://code.google.com/p/vt-middleware/wiki/vtldapSpring for more information.

<identifier>

The identifier of the group object is defined by a Grouper-specific Attribute Definition whose id is "group-dn" :

ldappc.xml	ldappc-resolver.xml
`<identifier id="group-dn" />`	`<resolver:AttributeDefinition id="group-dn" xsi:type="grouper:PSOIdentifier" structure="bushy" sourceAttributeID="extension" rdnAttributeName="cn" base="ou=groups,dc=example,dc=edu"> <resolver:Dependency ref="GroupDataConnector" /> </resolver:AttributeDefinition>`

<attribute>

The "cn" attribute of the group is provisioned with the value of the Grouper "name" attribute :

ldappc.xml	ldappc-resolver.xml
`<attribute name="cn" />`	`<resolver:AttributeDefinition id="cn" xsi:type="ad:Simple" sourceAttributeID="name" > <resolver:Dependency ref="GroupDataConnector" /> </resolver:AttributeDefinition>`

The "objectclass" attribute of the group is provisioned with the values "eduMember" and "groupOfNames" :

ldappc.xml	ldappc-resolver.xml

`<attribute name="objectclass" />`	`<resolver:AttributeDefinition id="objectclass" xsi:type="ad:Simple"> <resolver:Dependency ref="static" /> </resolver:AttributeDefinition>`
	
	`<resolver:DataConnector id="static" xsi:type="dc:Static"> <dc:Attribute id="objectclass"> <dc:Value>eduMember</dc:Value> <dc:Value>groupOfNames</dc:Value> </dc:Attribute> </resolver:DataConnector>`

To provision attributes based on membership, Grouper-specific Attribute Definitions are provided of the type "grouper:Member" which returns Member objects and "grouper:Group" which returns Group objects. The "sourceAttributeId" of these Attribute Definitions uses a special syntax to refer to memberships :

```
id = groups | members [ : all | effective | immediate | composite [ : fieldName ] ]
```

For example

```
id = "members"
```

is equivalent to

```
id = "members : all : members"
```

e.g. all of the members of the the default list, "members".

For every membership, the values of the provisioned attribute will be the values of the member's subject attribute as defined by <attribute> elements.

For example, the "hasMember" attribute of a group will be provisioned with the "lfname" of each member of the "jdbc" source and the "name" attribute of each member of the Grouper ("g:gsa") source :

ldappc.xml	ldappc-resolver.xml
<object id="group"> ... <attribute name="hasMember" id=" hasMember" /> ... </object>	<resolver:AttributeDefinition id="hasMember" xsi:type="grouper:Member" sourceAttributeID="members"> <resolver:Dependency ref="GroupDataConnector" /> <grouper:Attribute id="lfname" source="jdbc" /> <grouper:Attribute id="name" /> </resolver:AttributeDefinition>

To provision attributes based on Access privileges, the Grouper-specific "grouper:Subject" AttributeDefinition is provided. The sourceAttributeID of these Attribute Definitions also uses a special syntax :

```
id = admins | optins | optouts | readers | updaters | viewers [ : field ]
```

which is roughly equivalent to Group.getAdmins() etc.

ldappc.xml	ldappc-resolver.xml
<attribute name="admins" />	<resolver:AttributeDefinition id="admins" xsi:type="grouper: Subject"> <resolver:Dependency ref="groupDataConnector" /> <grouper:Attribute id="subjectId" source="jdbc" /> </resolver:AttributeDefinition>
<attribute name="adminOf" id="member- isAdminOf" />	<resolver:AttributeDefinition id="member-isAdminOf" xsi:type=" grouper:Group" sourceAttributeID="admins"> <resolver:Dependency ref="memberDataConnector" /> <grouper:Attribute id="name" /> </resolver:AttributeDefinition>

<reference>

A reference is similar to an attribute whose value is the identifier of another object. All <reference> elements require a "toPSOId" attribute whose value is the id of an <object> element.

In the following example, references to "group" and "member" objects will be provisioned as the "member" attribute. References to "member" objects consist of members whose source is "jdbc", and references to "group" objects consist of members whose source is "g:gsa" (Grouper).

ldappc.xml

```
<object id="group">
 ...
 <reference name="member" id="members-jdbc"
toPSOId="member" />
 <reference name="member" id="members-g:gsa"
toPSOId="group" />
 ...
</object>
```

ldappc-resolver.xml

```
<resolver:AttributeDefinition id="members-jdbc" xsi:
type="grouper:Member"
   sourceAttributeID="members">
  <resolver:Dependency ref="GroupDataConnector" />
  <grouper:Attribute id="subjectId" source="jdbc" />
</resolver:AttributeDefinition>

<resolver:AttributeDefinition id="members-g:gsa" xsi:
type="grouper:Member"
   sourceAttributeID="members">
  <resolver:Dependency ref="GroupDataConnector" />
  <grouper:Attribute id="name" source="g:gsa" />
</resolver:AttributeDefinition>
```

To provision an attribute whose values are the "name" attributes of the groups that a member belongs to, the Attribute Definition's sourceAttributeID will refer to the special attribute "groups", which uses the same membership syntax as "members" above :

ldappc.xml

```
<object id="member">
  ...
  <attribute name="isMemberOf" id="member-
isMemberOf" />
  ...
</object>
```

ldappc-resolver.xml

```
 <resolver:AttributeDefinition id="member-isMemberOf" xsi:
type="grouper:Group"
    sourceAttributeID="groups">
    <resolver:Dependency ref="MemberDataConnector" />
    <grouper:Attribute id="name" />
 </resolver:AttributeDefinition>
```

The reference is similar, and the values will be the identifiers of the groups that the member belongs to :

ldappc.xml

```
<object id="member">
  ...
  <reference name="memberOf" id="member-
isMemberOf" toPSOId="group"/>
  ...
</object>
```

ldappc-resolver.xml

```
 <resolver:AttributeDefinition id="member-isMemberOf"
xsi:type="grouper:Group"
    sourceAttributeID="groups">
    <resolver:Dependency ref="MemberDataConnector" />
    <grouper:Attribute id="name" />
 </resolver:AttributeDefinition>
```

Configuration Example : Multiple Targets: ldappc.xml

The following example provisions 4 targets, a production and test instance each of ActiveDirectory and OpenLDAP.

The `<targets>` element wraps `<target>` elements whose configurations are identical. The id of the `<targets>` element is for display purposes only.

To ease configuration, the value of `$target` will be rewritten with each <target/> elements "id" attribute.

Each <object> element must have a unique id, which is internally rewritten for Spring as "targetId:objectId".

```
<?xml version="1.0" encoding="utf-8"?>
<ldappc xmlns="http://grouper.internet2.edu/ldappc"
        xmlns:ldappc="http://grouper.internet2.edu/ldappc"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://grouper.internet2.edu/ldappc classpath:/schema/ldappc.xsd">

  <targets id="ActiveDirectory">

    <target id="ad-prod" provider="provider-ad-prod" />
    <target id="ad-test" provider="provider-ad-test" />

    <object id="group">
      <identifier id="groupDn-${target}" />
      <attribute name="objectclass" id="ad-objectclass" />
      <attribute name="cn" />
      <attribute name="description" />
      <attribute name="hasMember" />
      <attribute name="isMemberOf" id="groupIsMemberOf" />
      <reference name="member" id="members-jdbc" toObject="member" />
      <reference name="member" id="members-g:gsa" toObject="group" />
    </object>

    <object id="member">
      <identifier id="memberDn-${target}" />
    </object>

  </targets>

  <targets id="OpenLDAP" >

    <target id="openldap-prod" provider="provider-openldap-prod" />
    <target id="openldap-test" provider="provider-openldap-test" />

    <object id="group">
      <identifier id="groupDn-${target}" />
      <attribute name="objectclass" id="openldap-objectclass" />
      <attribute name="cn" />
      <attribute name="description" />
      <attribute name="hasMember" id="hasMember" />
      <attribute name="isMemberOf" id="groupIsMemberOf" />
      <reference name="member" id="members-jdbc" toObject="member" />
      <reference name="member" id="members-g:gsa" toObject="group" />
    </object>

    <object id="member">
      <identifier id="memberDn-${target}" />
      <attribute name="isMemberOf" id="memberIsMemberOf" />
      <reference name="memberOf" id="memberIsMemberOf" toObject="group" />
    </object>

  </targets>

</ldappc>
```

Configuration Example : Multiple Targets: ldappc-resolver.xml

Since we are using a single Attribute Resolver configuration file, identifiers for every object and for every target will need a unique id. Identifiers for group objects are calculated :

```
<!-- target specific identifiers -->

<resolver:AttributeDefinition id="groupDn-ad-prod" xsi:type="grouper:PSOIdentifier"
  structure="bushy" sourceAttributeID="extension" rdnAttributeName="cn" base="ou=groups,dc=example,dc=edu">
  <resolver:Dependency ref="GroupDataConnector" />
</resolver:AttributeDefinition>

<resolver:AttributeDefinition id="groupDn-ad-test" xsi:type="grouper:PSOIdentifier"
  structure="bushy" sourceAttributeID="extension" rdnAttributeName="cn" base="ou=groups,dc=test,dc=edu">
  <resolver:Dependency ref="GroupDataConnector" />
</resolver:AttributeDefinition>

<resolver:AttributeDefinition id="groupDn-openldap-prod" xsi:type="grouper:PSOIdentifier"
  structure="bushy" sourceAttributeID="extension" rdnAttributeName="cn" base="ou=groups,dc=example,dc=edu">
  <resolver:Dependency ref="GroupDataConnector" />
</resolver:AttributeDefinition>

<resolver:AttributeDefinition id="groupDn-openldap-test" xsi:type="grouper:PSOIdentifier"
  structure="bushy" sourceAttributeID="extension" rdnAttributeName="cn" base="ou=groups,dc=test,dc=edu">
  <resolver:Dependency ref="GroupDataConnector" />
</resolver:AttributeDefinition>
```

Identifiers for member objects are returned by SPMLDataConnectors, similar to LDAPDataConnectors, which execute searches. The "target" attribute refers to an SPML Provider Spring bean as defined in */dappc-services.xml*. The syntax of the SPML SearchRequest is currently unfinished, but will likely resemble LDAP filters for LDAP targets.

```
  <!-- ad-prod target identifier -->

  <resolver:AttributeDefinition id="memberDn-ad-prod" xsi:type="ad:Simple">
    <resolver:Dependency ref="ad-prod.DataConnector" />
  </resolver:AttributeDefinition>

  <resolver:DataConnector id="ad-prod.DataConnector" target="ad-prod" xsi:type="grouper:SPMLDataConnector"
    scope="subtree" base="ou=people,dc=example,dc=edu" returnData="identifier">
    <!--  SPMLSearchRequest goes here -->
  </resolver:DataConnector>

  <!-- ad-test target identifier -->

  <resolver:AttributeDefinition id="memberDn-ad-test" xsi:type="ad:Simple">
    <resolver:Dependency ref="ad-test.DataConnector" />
  </resolver:AttributeDefinition>

  <resolver:DataConnector id="ad-test.DataConnector" target="ad-test" xsi:type="grouper:SPMLDataConnector"
    scope="subtree" base="ou=people,dc=test,dc=edu" returnData="identifier">
    <!--  SPMLSearchRequest goes here -->
  </resolver:DataConnector>

  <!-- openldap-prod target identifier -->

  <resolver:AttributeDefinition id="memberDn-openldap-prod" xsi:type="ad:Simple">
    <resolver:Dependency ref="openldap-prod.DataConnector" />
  </resolver:AttributeDefinition>

  <resolver:DataConnector id="openldap-prod.DataConnector" target="openldap-prod" xsi:type="grouper:
SPMLDataConnector"
    scope="subtree" base="ou=people,dc=example,dc=edu" returnData="identifier">
    <!--  SPMLSearchRequest goes here -->
  </resolver:DataConnector>

  <!-- openldap-test target identifier -->

  <resolver:AttributeDefinition id="memberDn-openldap-test" xsi:type="ad:Simple">
    <resolver:Dependency ref="openldap-test.DataConnector" />
  </resolver:AttributeDefinition>

  <resolver:DataConnector id="openldap-test.DataConnector" target="openldap-test" xsi:type="grouper:
SPMLDataConnector"
    scope="subtree" base="ou=people,dc=test,dc=edu" returnData="identifier">
    <!--  SPMLSearchRequest goes here -->
  </resolver:DataConnector>

</AttributeResolver>
```

Configuration example : ldappc-services.xml

Derived from Shibboleth, services are defined in */ldappc-services.xml*.

```
<?xml version="1.0" encoding="UTF-8"?>

<Services xmlns="urn:mace:shibboleth:2.0:services"
          xmlns:attribute-afp="urn:mace:shibboleth:2.0:afp"
          xmlns:attribute-authority="urn:mace:shibboleth:2.0:attribute:authority"
          xmlns:attribute-resolver="urn:mace:shibboleth:2.0:resolver"
          xmlns:resource="urn:mace:shibboleth:2.0:resource"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xmlns:grouper="http://grouper.internet2.edu/shibboleth/2.0"
          xmlns:ldappc="http://grouper.internet2.edu/ldappc"
          xsi:schemaLocation="urn:mace:shibboleth:2.0:services classpath:/schema/shibboleth-2.0-services.xsd
                              urn:mace:shibboleth:2.0:afp classpath:/schema/shibboleth-2.0-afp.xsd
                              urn:mace:shibboleth:2.0:attribute:authority classpath:/schema/shibboleth-2.0-
attribute-authority.xsd
                              urn:mace:shibboleth:2.0:resolver classpath:/schema/shibboleth-2.0-attribute-
resolver.xsd
                              urn:mace:shibboleth:2.0:resource classpath:/schema/shibboleth-2.0-resource.xsd
                              http://grouper.internet2.edu/shibboleth/2.0 classpath:/schema/shibboleth-2.0-
grouper.xsd
                              http://grouper.internet2.edu/ldappc classpath:/schema/ldappc.xsd">
```

The ShibbolethAttributeResolver service :

```
<Service id="resolver" xsi:type="attribute-resolver:ShibbolethAttributeResolver">
  <ConfigurationResource file="ldappc-resolver.xml" xsi:type="resource:ClasspathResource" />
</Service>
```

The AttributeAuthority service :

```
<Service id="attribute-authority" xsi:type="grouper:SimpleAttributeAuthority" depends-on="resolver" resolver="
resolver" />
```

The ldappc service:

```
<Service id="ldappc" xsi:type="ldappc:ProvisioningServiceProvider" depends-on="attribute-authority"
    authority="attribute-authority">
  <ConfigurationResource file="ldappc.xml" xsi:type="resource:ClasspathResource" />
</Service>
```

Provider services for each target:

```
  <Service id="provider-ad-prod" xsi:type="ldappc:LdapPoolProvider" ldapPoolId="ldapPool">
    <ConfigurationResource file="ldappc-ad-prod.xml" xsi:type="resource:ClasspathResource" />
  </Service>

  <Service id="provider-ad-test" xsi:type="ldappc:LdapPoolProvider" ldapPoolId="ldapPool">
    <ConfigurationResource file="ldappc-ad-test.xml" xsi:type="resource:ClasspathResource" />
  </Service>

  <Service id="provider-openldap-prod" xsi:type="ldappc:LdapPoolProvider" ldapPoolId="ldapPool">
    <ConfigurationResource file="ldappc-openldap-prod.xml" xsi:type="resource:ClasspathResource" />
  </Service>

  <Service id="provider-openldap-test" xsi:type="ldappc:LdapPoolProvider" ldapPoolId="ldapPool">
    <ConfigurationResource file="ldappc-openldap-test.xml" xsi:type="resource:ClasspathResource" />
 </Service>

</Services>
```