LDAPPCNG

LDAPPCNG - LDAP Provisioning Connector New/Next Generation as of v1.6.0

> As of Grouper 2.1, LDAPPCNG has been replaced by the Provisioning Service Provider (PSP).

LDAPPCNG provisions group and membership information contained in the Groups Registry to an LDAP directory service.

Installation and usage information is on this page. Overview and general documentation, including an example, is here.

Install

Download the LDAPPCNG binary provisioning plugin for Grouper and expand it.

Copy the contents of the expanded package to your Grouper API directory. Configuration files are in the `conf` directory and java libraries are in `lib/custom`.

Usage

LDAPPCNG is run using GrouperShell (gsh).

For example, to maintain provisioning, polling every 60 seconds for changes :

```
bin/gsh.sh -ldappcng -bulkSync -interval 60
```

One of `-bulkCalc`, `-bulkDiff`, `-bulkSync`, `-calc <id>`, `-diff <id>`, or `-sync <id>` must be specified. All other arguments are optional.

Key	Value	Description
no arguments		Display usage.
-bulkCalc		Calculate provisioning for all identifiers.
-bulkDiff		Determine provisioning difference for all identifiers.
-bulkSync		Synchronize provisioning for all identifiers.
-calc <id>	*identifier*	Calculate provisioning for an identifier.
-diff <id>	*identifier*	Determine provisioning difference for an identifier.
-sync <id>	*identifier*	Synchronize provisioning for an identifier.
-entityName <id>	*entity identifier*	Provisioned object id. For example, group, member, etc.
-interval <seconds>	*seconds*	Number of seconds between the start of recurring provisioning iterations. If omitted, only one provisioning cycle is performed.
-lastModifyTime <id>	*yyyy-MM-dd[_hh:mm:ss]*	Select objects changed since this time.
-conf <dir>	*path to configuration files*	Configuration directory.
-logSpml		Log SPML requests and responses.
-output <file>	*file*	Print SPML responses to Output file. Default: STDOUT.
-printRequests		Print SPML requests as well as responses.
-requestID <id>	*request id*	SPML request identifier.
-returnData		Return data (identifier and attributes)
-returnEverything		Return everything (identifier, attributes, and references)
-returnIdentifier		Return identifier only.
-targetID <id>	*target id*	Target ID.

Configuration

Configuration files should be located on the Java classpath.

`ldappc-internal.xml`	Shibboleth Attribute Resolver
`ldappc-services.xml`	Shibboleth Attribute Resolver
`ldappc-resolver.xml`	Shibboleth Attribute Resolver
`ldappc-ldap.xml`	VT Ldap connector
`ldappcng.xml`	LDAPPCNG
`ldappc.properties`	Macro replacement

By default, macros of the form ${name} in `ldappcng.xml` will be replaced by their corresponding values in `ldappc.properties`.

Files prefixed with `ldappc` may also be used by ldappc.

ldappcng.xml

The `ldappcng.xml` file defines provisioned targets, objects, identifiers, attributes, and references.

<ldappc> - Provisioning Configuration

```
<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="LDAP">

    <target id="ldap" provider="ldap-provider" />

    <object id="stem">
      <identifier ref="stem-dn" baseId="${groupsOU}">
        <identifyingAttribute name="objectclass" value="organizationalUnit" />
      </identifier>
      <attribute name="objectClass" ref="stem-objectclass" />
      <attribute name="ou" ref="stem-ou" />
      <attribute name="description" ref="stem-description" />
    </object>

    <object id="group" authoritative="true">
      <identifier ref="group-dn" baseId="${groupsOU}">
        <identifyingAttribute name="objectClass" value="${groupObjectClass}" />
      </identifier>
      <attribute name="objectClass" ref="group-objectclass-eduMember" />
      <attribute name="cn" />
      <attribute name="description" />
      <attribute name="hasMember" ref="hasMember" />
      <attribute name="isMemberOf" ref="groupIsMemberOf" />
      <references name="member" emptyValue="" >
        <reference ref="members-jdbc" toObject="member" />
        <reference ref="members-g:gsa" toObject="group" />
      </references>
    </object>

    <object id="member">
      <identifier ref="member-dn" baseId="${peopleOU}">
        <identifyingAttribute name="objectclass" value="person" />
      </identifier>
      <attribute name="objectClass" ref="member-objectclass" retainAll="true" />
      <attribute name="isMemberOf" ref="memberIsMemberOf" />
    </object>

  </targets>

</ldappc>
```

<targets>

The targets element allows more than one target to be provisioned using the same configuration. This may be useful, for example, when provisioning a production and test environment identically.

```
<targets id="ID" >
  <target ...
  <target ...
    <object ...
</targets>
```

attribute	description
id	Uniquely identifies a collection of targets.

<target>

A target contains objects. Each target requires a unique identifier and a provider identifier. Multiple target elements are allowed.

```
<target id="ID" provider="providerID" />
```

attribute	description
id	Unique identifier.
provider	Identifier of a provider defined in the attribute resolver services configuration.

For example, LDAPPCNG ships with an LDAP provider using the vt-ldap distribution.

ldappcng.xml

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

ldappc-services.xml

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

<object>

A provisioned object. For example, a group, member, stem, account, etc. An object consists of an identifier, attributes, and references.

```
<object id="ID">
  <identifier ...
  <attribute ...
  <references ...
</object>
```

attribute	description
id	Uniquely identifies the object per target.

<identifier>

All objects require a unique identifier. The value of the identifier is returned from the Shibboleth Attribute Resolver.

```
<identifier ref="REF" baseId="BASE">
  <identifyingAttribute ...
</identifier>
```

attribute	description
ref	The id of an attribute definition defined in the attribute resolver configuration.
baseId	The identifier of the container (the SPML2 containerID).

<identifyingAttribute>

This element maps an object returned from a target provider to an object in the LDAPPCNG configuration. This is not specified anywhere in the SPML specification and is likely a candidate for improvement.

```
<identifyingAttribute name="NAME" value="VALUE" />
```

attribute	description

name	Attribute name.
value	Attribute value.

For example, an object returned from a target which has an attribute named "objectclass" with value "groupOfNames" will be identified as a "group" object.

ldappc-services.xml

```
<object id="group">
  <identifier ref="group-dn" baseId="ou=groups,dc=example,dc=edu">
    <identifyingAttribute name="objectClass" value="groupOfNames" />
  </identifier>
```

<attribute>

A provisioned attribute. The value of the attribute is returned from the Shibboleth Attribute Resolver.

```
<attribute name="NAME" ref="REF" />
```

attribute	description
name	The name of the provisioned attribute.
ref	The id of an attribute definition defined in the attribute resolver configuration.

<references>

Defines references to other objects.

```
<references name="NAME" emptyValue="" >
  <reference ... />
</references
```

attribute	description
name	The provisioned attribute name.
empty-value	Optional. Defines the value of the provisioned attribute if no references are returned from the attribute resolver. This should be defined when provisioning a required (MUST) ldap attribute, such as "member" of an OpenLDAP directory.

<reference>

Defines a reference to another object. The value is

```
<reference ref="REF" toObject="OBJECTID" /
```

attribute	description
ref	The id of an attribute definition defined in the attribute resolver configuration.
toObject	The id of the Provisioned Object referred to.

For example, the following configuration will return references to the identifiers of "member" objects for the attribute definition "members-jdbc", and references to the identifiers of "group" objects for the attribute definition "member-g:gsa".

The "members-jdbc" attribute's values will consist of the "id" attribute for every subject which is a member of a group's "members" attribute.

The "members-g:gsa" attribute's values will consist of the "name" attribute for every group which is a member of a group's "members" attribute.

The values of the "members-jdbc" and "members-g:gsa" attributes are passed to the attribute resolver to determine their identifiers.

ldappcng.xml

```
<references name="member" emptyValue="" >
  <reference ref="members-jdbc" toObject="member" />
  <reference ref="members-g:gsa" toObject="group" />
 </references>
```

ldappc-resolver.xml

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

Example: calc

To print to STDOUT the SPML representation of how an object should be provisioned :

```
>bin/gsh.sh -ldappcng -calc stem:groupName

<ldappc:calcResponse status='success' requestID='2010...QKUSL7CS' ... >
  <ldappc:id ID='stem:groupName'/>
  <ldappc:pso entityName='group'>
    <psoID ID='cn=stem:groupName,ou=groups,dc=example,dc=edu' targetID='ldap'/>
    <data>
      <dsml:attr name='objectClass' ... >
        <dsml:value>top</dsml:value>
        <dsml:value>groupOfNames</dsml:value>
        <dsml:value>eduMember</dsml:value>
      </dsml:attr>
      <dsml:attr name='cn' ... >
        <dsml:value>groupName</dsml:value>
      </dsml:attr>
      <dsml:attr name='hasMember' ... >
        <dsml:value>member1</dsml:value>
        <dsml:value>member2</dsml:value>
      </dsml:attr>
      <dsml:attr name='isMemberOf' ... >
        <dsml:value>stem:otherGroup</dsml:value>
      </dsml:attr>
    </data>
    <capabilityData mustUnderstand='true' capabilityURI='urn:oasis:names:tc:SPML:2:0:reference'>
      <spmlref:reference typeOfReference='member' ... >
        <spmlref:toPsoID ID='cn=member1,ou=people,dc=example,dc=edu' targetID='ldap'/>
      </spmlref:reference>
      <spmlref:reference typeOfReference='member' ... >
        <spmlref:toPsoID ID='cn=member2,ou=people,dc=example,dc=edu' targetID='ldap'/>
      </spmlref:reference>
    </capabilityData>
  </ldappc:pso>
</ldappc:calcResponse>
```

Example: diff

To print to STDOUT the SPML representation of changes that should be made :

```
>bin/gsh.sh -ldappcng -diff stem:groupName

<ldappc:diffResponse status='success' requestID='2010..._QKUSQLQ0' ... >
  <modifyRequest entityName='group' requestID='2010..._QKUSQLRM' returnData='everything' ... >
    <psoID ID='cn=um:manual:g20031124220052001,ou=groups,dc=memphis,dc=edu' targetID='ldap'/>
    <modification modificationMode='add'>
      <dsml:modification name='description' operation='add' ...>
        <dsml:value>A Description</dsml:value>
      </dsml:modification>
    </modification>
  </modifyRequest>
  <ldappc:id ID='stem:groupName'/>
</ldappc:diffResponse>
```