Active Directory (LDAP) Setup

The following are some notes on setting up Apache DS with Directory Studio:

1. Setup ApacheDS

Start apacheds: (on Ubuntu)

sudo service apacheds-2.0.0-M17-default start

From Apache Directory Studio

New Connection port 10389

Bind DN or user: uid=admin,ou=system
Bind password: secret

2. Add a new partition

Click the connection and choose Open Connection.

Click "Advanced partitions configuration".

Add partition ID=elixirbase Suffix=dc=elixirbase,dc=com

Save changes (Eclipse Save button).

Restart apacheds:

sudo service apacheds-2.0.0-M17-default restart

3. Refresh tree and delete the node dc=elixirbase,dc=com because the ldif includes it.

Use **File-> Import -> LDIF** into LDAP to load elixirbase.ldif

(The LDIF file is in the misc directory of the Elixir Ambience build.)

Now we have the elixirbase ldif loaded into ApacheDS for queries.

4. The first thing to change is the run- script, so copy run-server.sh into run-ldap.sh

5. Add the following to the java startup:

-Delixir.usergroup.factory=com.elixirtech.arch.usergroup.ldap.LDAPUserGroupDomainFactory

You can also put this in the *etc/application.conf* if you prefer (without the -D)

6. Make sure your apacheds service is running and launch Ambience using *run-ldap.sh*

7. Now logon to the DomainManager.

If your LDAP server is running on a different host or port, there will be an error in the log. Visit **Configuration -> module -> usergroupdb -> ldap** and alter the host and port values accordingly.

8. Next visit **Configuration -> module -> usergroupdb -> function -> SignIn**.

Edit the users array at the end of the config to include aimi (an administrator):

"users":["aimi"]

You will need to restart the server after saving these changes.

Note: Users david and aimi (admin user) referred below are hypothetical users just used for documentation. Use the actual usernames that are relevant to your configuration.

9. Now you can logon to Ambience as aimi (password aimi).

10. The next step after logging in as aimi is to visit

Admin -> Access Matrix.

Here you can turn on SignIn permissions for other users (subject to licence limits). Turn on access for david, then logoff and verify you can logon as david (password david). David is not an admin user (those in group Atlas are admin users), so you will need to logon as aimi to make further admin changes.

10. Finally as aimi, you should move the contents of /User/admin/cron into /User/aimi/cron, otherwise the scheduler will attempt to run the jobs as admin when that user exists in LDAP. You can then delete the default admin and test user folders.

11. Once you have gone through these steps to verify your LDAP system is working properly, you use the domain manager to look into the LDAP configuration.

First visit **Configuration -> module -> usergroupdb -> ldap**

The config here reads:

{"version":1,

"connectionFactory":"com.elixirtech.arch.usergroup.ldap.DefaultLDAPConnectionFactory"
,

"cxtFactory":"com.sun.jndi.ldap.LdapCtxFactory",

"host":"localhost",

"port":10389,

"protocol":"default",

"method":"simple",

"user":"uid=admin,ou=system",

"passwordEncrypted":"8Oz2e9XJ+Grd396E4QK91Q=="}

You might need to edit host and port to connect to a different server.

The user/password here represents LDAP credentials to read all users and groups.

12. There are several sub-configurations for the DefaultLDAPConnectorFactory (other custom connector factories may vary). Look at the contents of the sample ldif to help understand these.

users contains:

{"name":"ou=users,dc=elixirbase,dc=com","mode":"one-level","returnAttr":"uid","focus":{"left":"uid"}}

This indicates where the user information is stored in LDAP - one level below ou=users,dc=elixirbase,dc=com. The user id is stored as uid and we want to focus on this uid to get our set of user names.

groups contains:

{"name":"ou=groups,dc=elixirbase,dc=com","mode":"one-level","returnAttr":"roleOccupant","focus":{"left":"cn","right":"uid"}}

This is similar to users, but now we want to not only get the groups, but the users assigned to those groups. In a Map structure this is [Group-> List[User]].

The groups are found one-level below ou=groups,dc=elixirbase,dc=com and the users are found in the roleOccupant attributes. We then focus our attention on the cn (which is the group name) and the uid (which is the user name). This gives us [cn-> List[uid]] (note the left and right positions).

Finally adminGroups contains:

{"name":"ou=groups,dc=elixirbase,dc=com","mode":"subtree","filter":"cn=Atlas","returnAttr":"cn","focus":{"left":"cn"}}

This filters all the groups, selecting the one whose name is "Atlas" and returning the cn of that group. This makes Atlas the admin group and anyone who is a member of Atlas is considered an administrator.

This sample represents a typical layout of users and groups in LDAP, however some systems follow different structures.

If the particular LDAP cannot be modeled using these three query structures, then a custom LDAP mechanism can be included by altering:

"connectionFactory":"com.elixirtech.arch.usergroup.ldap.DefaultLDAPConnectionFactory"

to point to a custom implementation as Elixir Professional Services.