Posixldapauth plugin - Manual

Overview

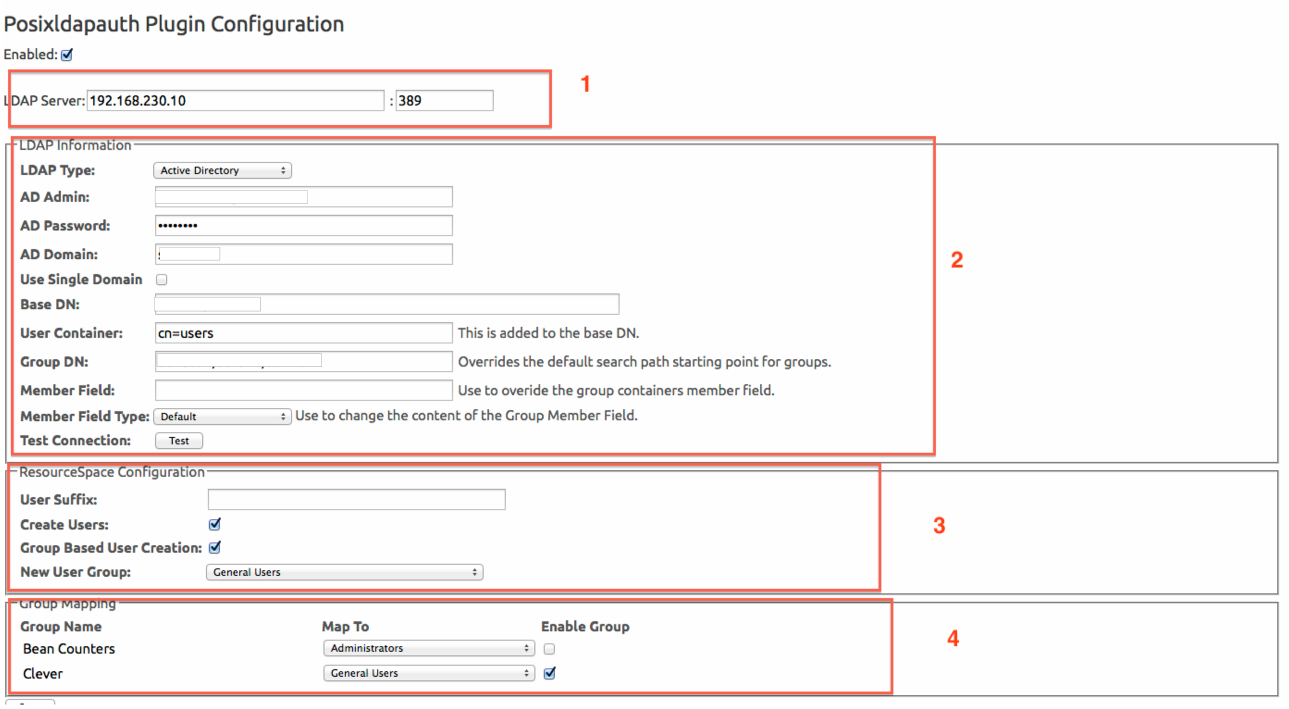
This plugin is designed to use an external LDAP for authentication to ResourceSpace.

It has been tested with Active Directory, Open Directory on OSX Server and OpenLDAP.

The original goal was to make the integration as simple and as flexible as possible, however the plugin can be quite complicated to setup depending on your directory structure.

Options Page

The options page is divided into a number of key areas.



Posixldapauth Plugin Configuration

Enabled: ☑

LDAP Server: 192.168.230.10 : 389

LDAP Information

LDAP Type: Active Directory
AD Admin: [redacted]
AD Password: ••••••••
AD Domain: [redacted]
Use Single Domain □
Base DN: [redacted]
User Container: cn=users This is added to the base DN.
Group DN: [redacted] Overrides the default search path starting point for groups.
Member Field: Use to overide the group containers member field.
Member Field Type: Default Use to change the content of the Group Member Field.
Test Connection: Test

ResourceSpace Configuration

User Suffix:
Create Users: ☑
Group Based User Creation: ☑
New User Group: General Users

Group Mapping

Group Name	Map To	Enable Group
Bean Counters	Administrators	□
Clever	General Users	☑

Section 1 LDAP Server Address and Port:

Use this to specify the IP Address or DNS name of the LDAP Server, and the communication port. Default port is 389.

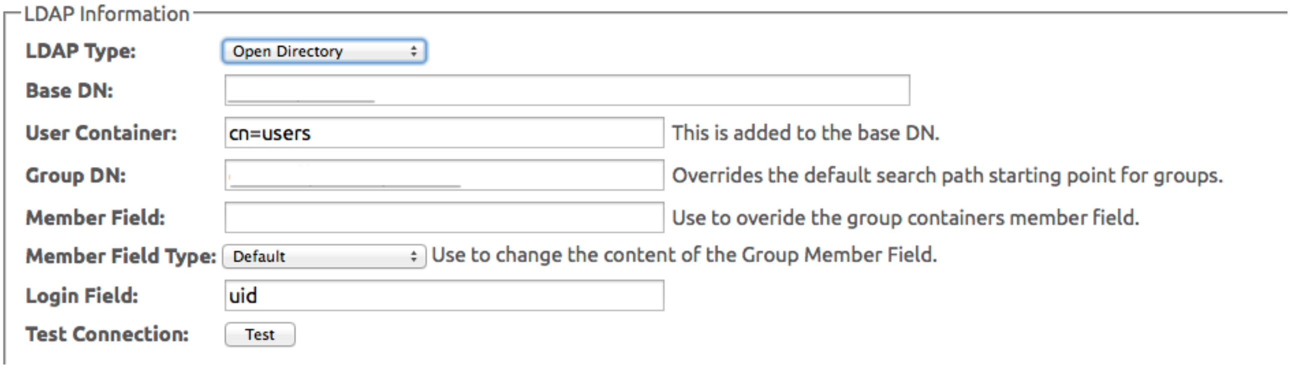
Section 2: LDAP Information

This is split into two sets of config options depending on the LDAP type selected from the drop down box.

Although they both have shared fields, the behaviour of the shared fields is different for each type of directory.

Open Directory

This is used for Apple Open Directory and OpenLDAP.



LDAP Information

LDAP Type: Open Directory

Base DN:

User Container: cn=users This is added to the base DN.

Group DN: Overrides the default search path starting point for groups.

Member Field: Use to overide the group containers member field.

Member Field Type: Default Use to change the content of the Group Member Field.

Login Field: uid

Test Connection: Test

Base DN:

This is the base or root DN of the LDAP. e.g. example.com

User Container:

This is where the users are stored, it’s added to the Base DN, so for example cn=users becomes cn=users,dc=example,dc=com

Group DN:

Putting an entry in this field overrides the default OU that the plugin looks for groups in the LDAP. The default is mapped for Apple Open Directory which is cn=groups,dc=example,dc=com. An entry in this field must be the full DN.

Member Field:

Putting an entry in this field will override the default entry in the Group object that plugin looks for to find the list of users. The default is memberuid.

Member Field Type:

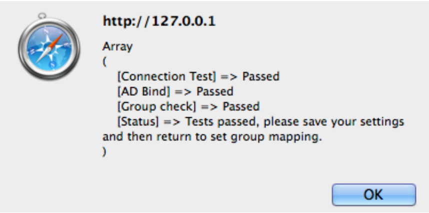
This selects the how the users are stored in the Member Field above. The default is userName, i.e. the users short name.

Login Field:

This is a legacy field and should be left as “uid”.

Test Button

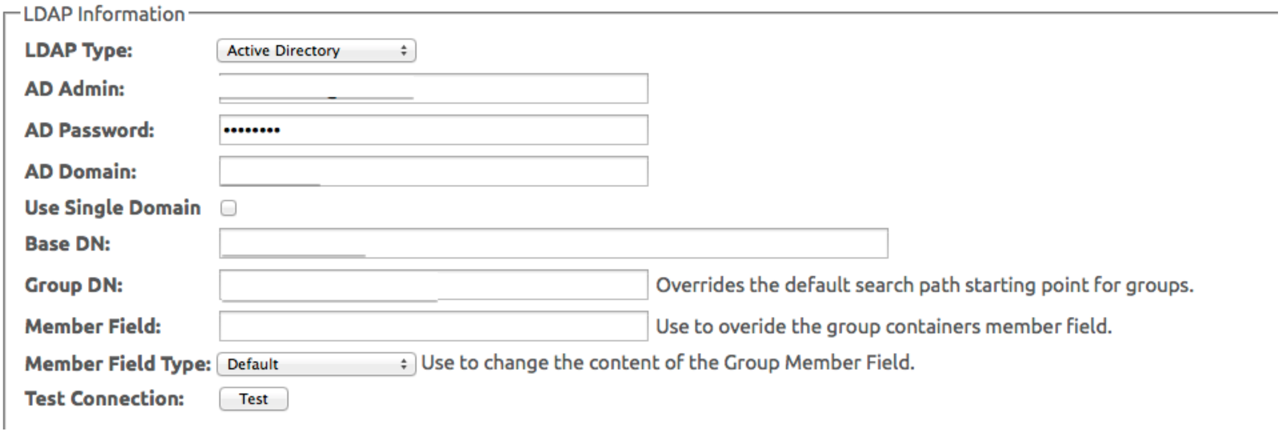
This will test the settings by attempting to authenticate to the directory and retrieve a list of groups.



http://127.0.0.1

```
Array
(
    [Connection Test] => Passed
    [AD Bind] => Passed
    [Group check] => Passed
    [Status] => Tests passed, please save your settings
and then return to set group mapping.
)
```

Active Directory



LDAP Information

LDAP Type:	Active Directory	
AD Admin:		
AD Password:	••••••••	
AD Domain:		
Use Single Domain	□	
Base DN:		
Group DN:		Overrides the default search path starting point for groups.
Member Field:		Use to overide the group containers member field.
Member Field Type:	Default	Use to change the content of the Group Member Field.
Test Connection:	Test	

AD Admin
The AD admin account to to bind to the AD with in order to get the groups list etc. the format is admin@example.com

AD Password
The password for the above account.

AD Domain:
The base domain eg example.com

Use Single Domain:
Ticking the box will append the domain name to user accounts when they login so they don’t have to type the domain name. ie they can login with peterpan rather than peterpan@example.com.

Base DN:
This is the base DN of the domain, i.e. dc=example, dc=com.

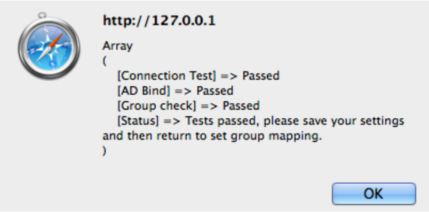
Group DN:
Putting an entry in this field overrides the default place that the plugin starts the search downwards for groups in the AD. The default behaviour is to search for all groups from the base DN down. An entry in this field must be the full DN to start the search from.

Member Field:
Putting an entry in this field will override the default entry in the Group object that plugin looks for to find the list of users. The default is member.

Member Field Type:
This selects the how the users are stored in the Member Field above. The default is RDN, i.e. cn=peterpab,ou=staff,dc=example,dc=com.

Test Button

This will test the settings by attempting to authenticate to the directory and retrieve a list of groups.

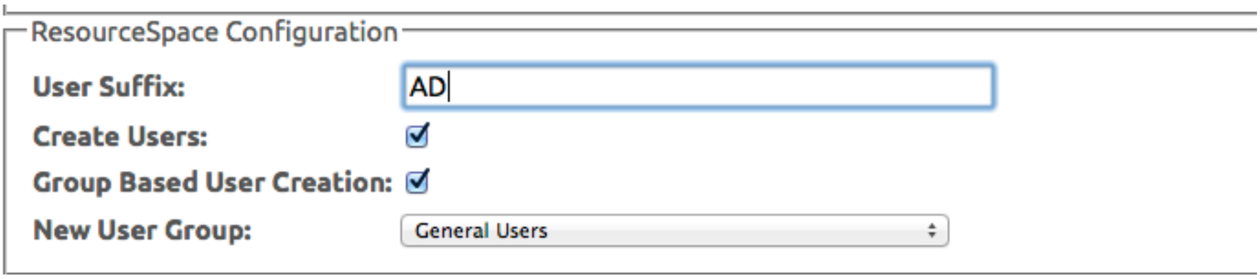


http://127.0.0.1

```
Array
(
    [Connection Test] => Passed
    [AD Bind] => Passed
    [Group check] => Passed
    [Status] => Tests passed, please save your settings
and then return to set group mapping.
)
```

Section 3 : ResourceSpace Configuration

This section specifies what RS will do with the authenticated accounts.



ResourceSpace Configuration

User Suffix: AD

Create Users: ☑

Group Based User Creation: ☑

New User Group: General Users

User Suffix

This is appended to the user account name in RS to denote that it is an external account.

Create Users

This will create mirrored user accounts in RS. It is currently necessary to do this as the user will not be able to log in unless an account is in the RS database.

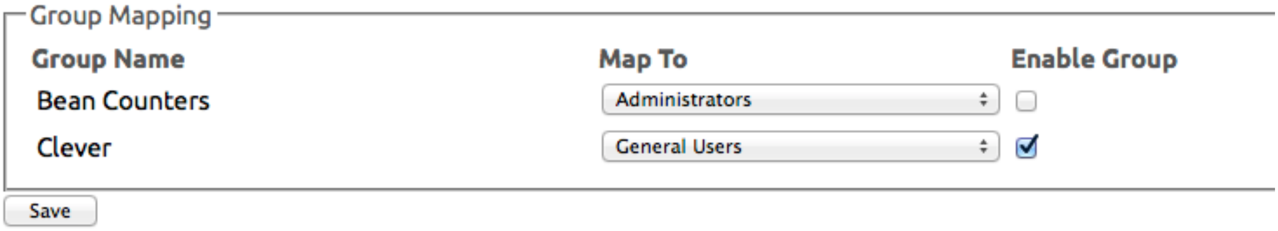
Group Based User Creation

This enables the group mapping that allows groups in the directory to be mapped to roles in RS.

New User Group

If group based creation is not enabled, all authenticated users will become members of this group.

Section 4: Group Mapping



Group Mapping

Group Name	Map To	Enable Group
Bean Counters	Administrators	□
Clever	General Users	☑

This allows you to map groups in the directory to roles in ResourceSpace and enable the groups you want to be able to log in.

If a user is a member of multiple groups and they are mapped to different roles, the user will get the mapping that has the highest privileges.

Known issues:

1.Changes are not being saved after the initial setup

In a directory with more than 400 groups it is likely that the php system variable max_input_vars is being exceeded. To fix this add the following to your php.ini file and restart the webserver.

```
max_input_vars=10000
```

2. Directories with more than 1000 groups

This will currently prevent the plugin from working. The only workaround is to use the Group DN field to start the group search further down the tree so that a smaller number of groups are returned.