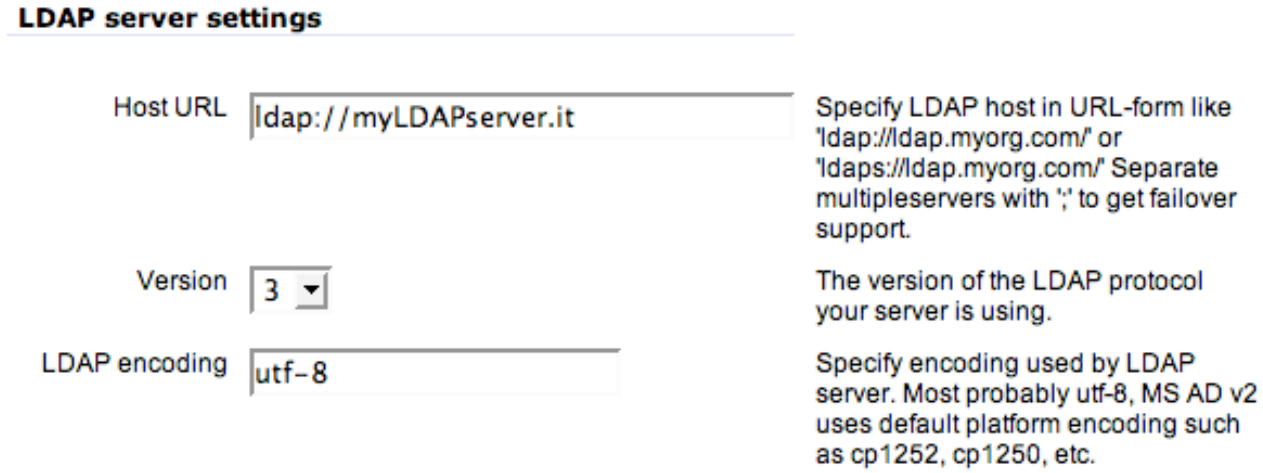
Steps to setup authentication and enrolment through LDAP protocol

Step 1: Authentication

The web user try to get inside Moodle. Moodle will recognize him/her only if his credentials are found inside Accounts stored in the context explained to Moodle.

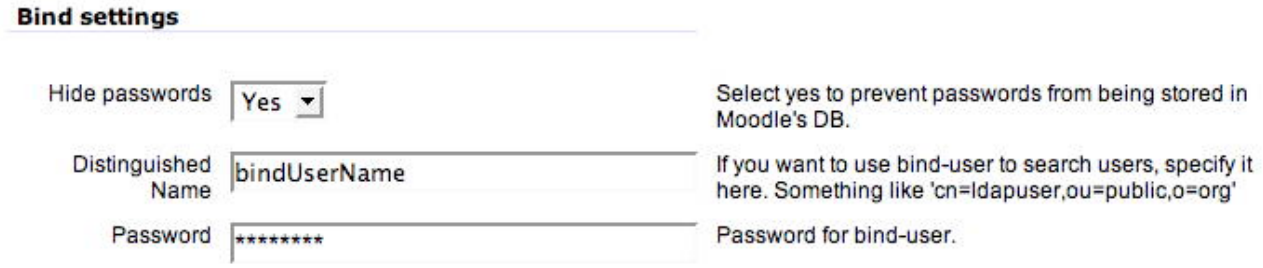
To do this it is necessary to tell Moodle

1. Where the LDAP db is.



Picture 1

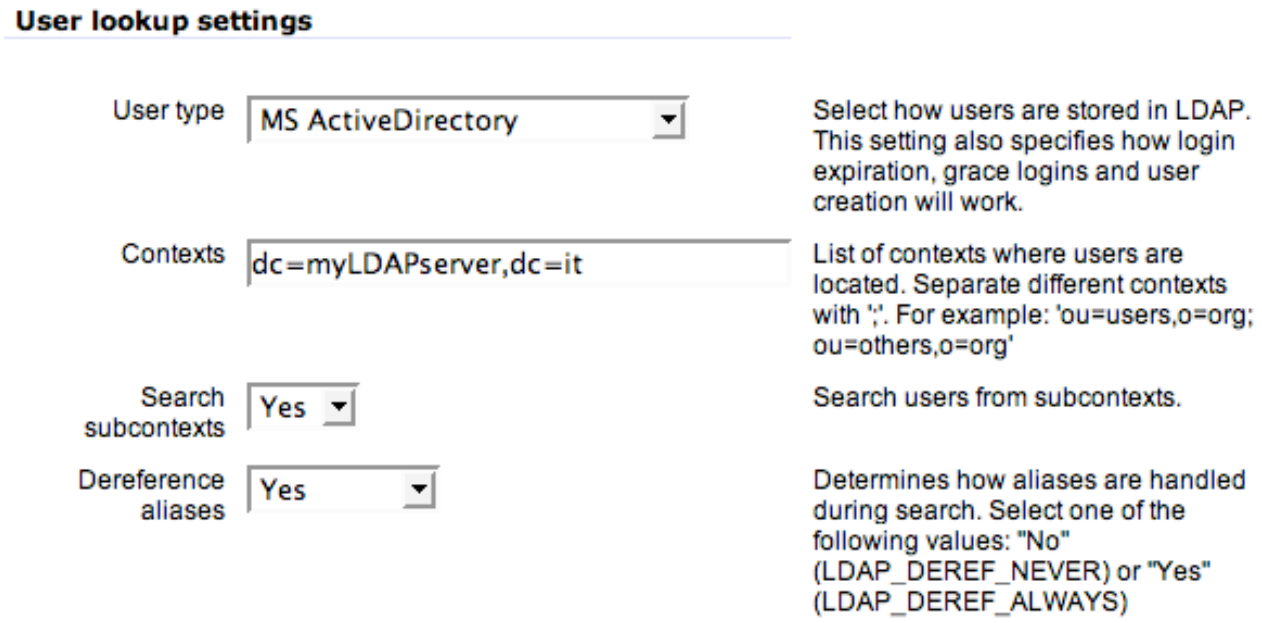
2. Which user is operating in LDAP



Picture 2

This user has ho have search right into the context/s where accounts are stored

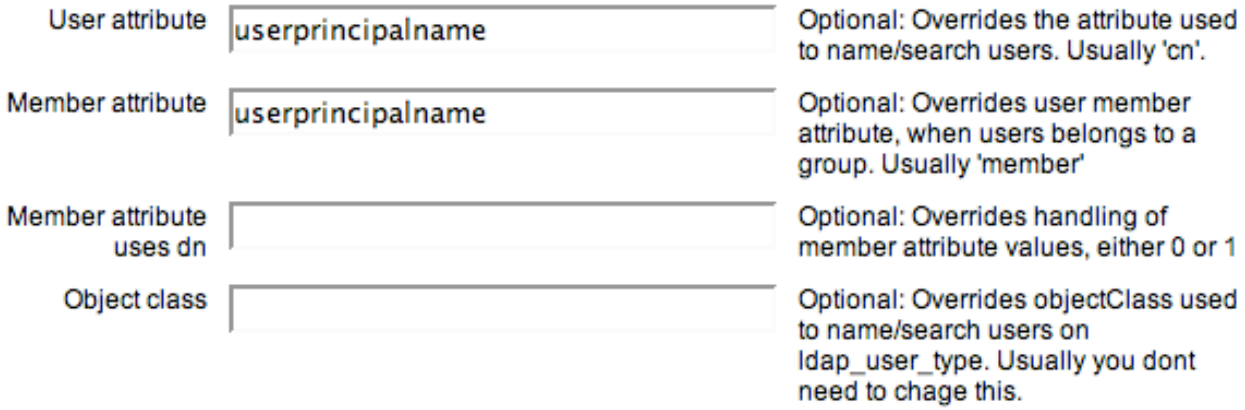
3. How Accounts of user are stored and where they are stored



Picture 3

If **search subcontexts** is set to Yes, your users account can be wherever inside the declared context.
Deference aliases is a Boolean variable. It defines if eventually aliases are considered as an object (dereference == no) or if is the object pointed by the alias the one you wan to consider (dereference == yes).

4. Where username and password are stored



Picture 4

User attribute is the name of the field that authentication process will use to look for the user username in order to authenticate him/her.

Some usual questions.
Question01: Where am I supposed to enter the name of the field in which LDAP stores users password?
Answer01: nowhere. You don't need it, because Moodle never 'reads' the password from the LDAP server. In fact, you can't read it with 99'99% of the LDAP servers out there. Moodle checks the username/password combination entered by the web user, by trying to bind to the LDAP server with that combination and checking the return code/error. If no error is returned, this means that user who tried that username/password combination is allowed to get inside.

Question02: What is Member attribute for?
Answer02: When we are checking membership in a group, we need to read the value of a certain attribute of the group object (not the user object). Different LDAP servers use different membership attributes, so we need to specify the right name here. If you don't specify anything here, Moodle will choose a default value depending on the 'User type' you specified above (which is usually right).

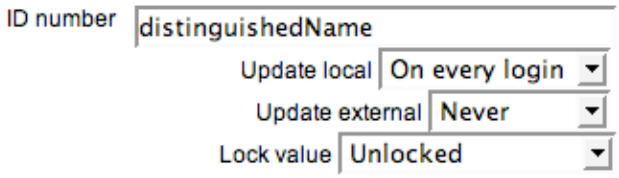
Question03: What is Object class for?
Answer03: Every single object in a LDAP server has an object class (a type): user, group, organizationaUnit (ou), etc. Different LDAP servers use different object classes to store user accounts. So, much like the membership attribute, you should specify the right user object class for your LDAP server. If you don't, Moodle again uses a default object class value depending on the 'User type' you specified above.

According to what already written, the authentication process is here described:
Moodle searches the LDAP directory to find an object of the right object class type ('Object class' setting above) with a value in its user attribute ('User attribute' setting above) equal to the value entered in the username login page. If it finds it (gets its distinguished name, i.e., the path to the user object inside the LDAP server), it tries to bind to the LDAP server using the distinguished name of the user and the password entered in the login page. If the password is good, binding will succeed and Moodle will let the user in. If Moodle can't find the user or the binding fails, it will flag the error and redirect the user to the login page again.

Data Mapping 1 of 2
If authentication passes Moode will create a local account for the LDAP user. It is to trace the user. If mandatory fields of the just authenticated user are not defined, Moodle will prompt with the edit user profile page. This in order to get all mandatory fields. If you don’t want this web page to appear, map, at least:

- First name
- Surname
- Email address

Data Mapping 2 of 2
If authentication passes its ID number will distinguish the user to match his/her account during LDAP enrolment. This ID number is defined in:



Picture 5

All the other options are not mandatory and I’ll neglect them, up to now.

Rationale.
Inside LDAP the administrator has to create two conceptually different environments.
The first is for the authentication, the latter is for enrolment.

inside declared context (eventually somewhere in subcontexts) admin has to store accounts for users that will be able to authenticate

Accounts inside LDAP directory for authentication

Still into LDAP, but in a different position, admin has to create a structure of ou ending with groups. The structure has to reflect each Moodle course's short name in which admin wants to enrol from LDAP. Per each course, admin has to provide an ou with the list of allowed roles.
For instance:

GROUPS of accounts inside LDAP. Moodle like structures of courses for enrolment

Ou= Moodle
Ou= Teacher
group= Course01
account= accountUserA
account= accountUserB
account= accountUserC
group= Course02
account= accountUserA
account= accountUserD
account= accountUserE
Ou= Student
group= Course01
account= accountUserF
account= accountUserG
group= Course02
account= accountUserG
account= accountUserH
account= accountUserI

Step 2: Enrolment

If authentication passes, Moodle will use the 'ID Number' value of the authenticated user (taken from the first part of LDAP structure, see Data Mapping 2 of 2) in order to find him/her in the second part of LDAP structure.
Moodle now only knows the authenticated user and recognize him/her from his/her ID Number.

The authenticated user now tries to get inside a course.
Question04: What do Moodle do in order to allow/deny him/her the access to the course?
Answer04: Before answering this question let me write down an LDAP generic structure. The structure shown below is shortly defined in red in the next few lines

```
moodle
--student
----course01
------user01
------user02
----course02
------user03
```

```
--teacher
----course01
------user03
```

Before the LDAP structure of courses, a generic user account description will be shown.

Generic user account description:

```
1  ************************************
2  ** Structure of generis user account **
3  ************************************
4  dn: CN=user01FistName.user01LastName,CN=moodle,DC=myInstitute,DC=it
5  changetype: add
6  objectclass: top
7  objectclass: person
8  objectclass: organizationalPerson
9  objectclass: user
10 cn: user01FistName.user01LastName
11 sn: user01LastName
12 givenName: user01FistName
13 distinguishedName: CN=user01FistName.user01LastName,OU=moodle,DC=myInstitute,DC=it
14 instanceType: 4
15 ..some more user record details..
16 ******************************************
17 ** End of structure of generis user account **
18 ******************************************
```

MSAD LDAP structure export

```
19 ****************************
20 ** Structure of ou=moodle **
21 ****************************
22 dn: OU=moodle,DC=myInstitute,DC=it
23 changetype: add
24 objectClass: top
25 objectClass: organizationalUnit
26 ou: moodle
27 distinguishedName: OU=moodle,DC=myInstitute,DC=it
28 ..some more ou details..
29 **********************************
30 ** End of structure of ou=moodle **
31 **********************************
32
33 ****************************
34 ** Structure of ou=Student **
35 ****************************
36 dn: OU=Student,OU=moodle,DC=myInstitute,DC=it
37 changetype: add
38 objectClass: top
39 objectClass: organizationalUnit
40 ou: Student
41 distinguishedName: OU=Student,OU=moodle,DC=myInstitute,DC=it
42 ..some more ou details..
43 **********************************
44 ** End of structure of ou=Student **
45 **********************************
46
47 ********************************************************************************
48 ** Structure of cn=course01 (inside ou=Student) with one belonging user **
49 ********************************************************************************
50 dn: CN=course01,OU=Student,OU=moodle,DC=myInstitute,DC=it
51 changetype: add
52 objectClass: top
53 objectClass: group
54 cn: course01
55 member: CN=user01FistName.user01LastName,CN=Users,DC=myInstitute,DC=it
56 member: CN=user02FistName.user02LastName,CN=Users,DC=myInstitute,DC=it
57 distinguishedName: CN=course01,OU=Student,OU=moodle,DC=myInstitute,DC=it
58 ..some more cn details..
59 ********************************************************************************
60 ** End of structure of cn=course01 (inside ou=Student) with one belonging user **
61 ********************************************************************************
62
63 ********************************************************************************
64 ** Structure of cn=course02 (inside ou=Student) with one belonging user **
65 ********************************************************************************
66 dn: CN=course02,OU=Student,OU=moodle,DC=myInstitute,DC=it
67 changetype: add
68 objectClass: top
```

```
 69 objectClass: group
 70 cn: demo
 71 member: CN=user03FistName.user03LastName,CN=Users,DC=myInstitute,DC=it
 72 distinguishedName: CN=course02,OU=Student,OU=moodle,DC=myInstitute,DC=it
 73 ..some more cn details..
 74 ********************************************************************************
 75 ** End of structure of cn=course02 (inside ou=Student) with one belonging user **
 76 ********************************************************************************
 77
 78 ****************************
 79 ** Structure of ou=Teacher **
 80 ****************************
 81 dn: OU=Teacher,OU=moodle,DC=myInstitute,DC=it
 82 changetype: add
 83 objectClass: top
 84 objectClass: organizationalUnit
 85 ou: Teacher
 86 distinguishedName: OU=Teacher,OU=moodle,DC=myInstitute,DC=it
 87 ..some more ou details..
 88 **********************************
 89 ** End of structure of ou=Teacher **
 90 **********************************
 91
 92 ************************************************************************
 93 ** Structure of cn=course01 (inside ou=Teacher) with one belonging user **
 94 ************************************************************************
 95 dn: CN=course01,OU=Teacher,OU=moodle,DC=myInstitute,DC=it
 96 changetype: add
 97 objectClass: top
 98 objectClass: group
 99 cn: course01
100 member: CN=user03FistName.user03LastName,CN=Users,DC=myInstitute,DC=it
101 distinguishedName: CN=course01,OU=Teacher,OU=moodle,DC=myInstitute,DC=it
102 ..some more cn details..
103 ********************************************************************************
104 ** End of structure of cn=course01 (inside ou=Teacher) with one belonging user **
105 ********************************************************************************
```

Moodle admin said to Moodle that the user will be uniquely defined by its `distinguishedName` defined in row 13 (see pictures 5). So Moodle starts to look for this information wherever it may be in the second part of the LDAP interface.

As it can be easily seen the `distinguishedName` of the line 13 (`CN=user01FistName.user01LastName,OU=moodle,DC=myInstitute,DC=it`) can be found in line 55 inside of the `uo=moodle` (see lines: 19..31) `uo=student` (see lines: 33..45), `cn=course01` (47..61)
This means that while Moodle will start to look in each LDAP object, Moodle admin has to tell Moodle to look inside `objectClass: group` (see row 53), in the field `member` (see row 55).

Once found, Moodle will be ready to enrol the user found in line 55.

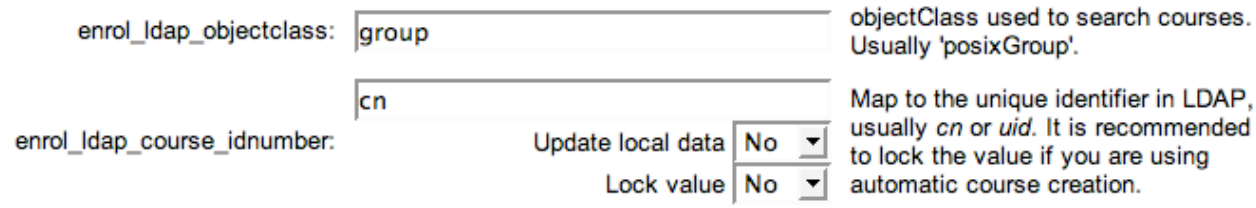
What is still missing is the name of the Moodle course in which the user has to be enrolled. Well Moodle admin has to provide the object containing the name of the course. In other words, Moodle admin has to tell Moodle that the name of the course is in line 54. So, it has to be specified that the object containing the course name is “cn”
This is done by defining the “enrol_LDAP_course_idnumber” variable.

What follow is the Moodle admin interface to define objectClass `(group)`, LDAP member attribute `(member)` and enrol_LDAP_course_idnumber `(cn)`. You can find it in Site administration -> Courses -> Enrolments

Role mapping

Roles	LDAP contexts	LDAP member attribute
Administrator		
Course creator		
Teacher	OU=Teacher,OU=moodle,DC=myInstitu	member
Non-editing teacher		
Student	OU=Student,OU=moodle,DC=myInstitu	member
Guest		
Authenticated user		

Course enrolment settings



So Moodle will searches for LDAP objects whose

1. LDAP ‘member’ attribute value who contains the user 'ID Number' value
2. whose objectClass equals the value of 'enrol_ldap_objectclass'.

That is, if a user 'ID Number' is `CN=user01FistName.user01LastName,OU=moodle,DC=myInstitute,DC=it`, 'LDAP member attribute' is `member`, 'LDAP Contexts' is `ou=students,ou=moodle,dc=myInstitute,DC=org'` and 'enrol_ldap_objectclass' is `group` Moodle will search for LDAP objects inside that context whose member attribute contains the value `CN=user01FistName.user01LastName,OU=moodle,DC=myInstitute,DC=it` and whose objectClass attribute value is `group` and it will find the user described in the row 55. That user will be allowed to get inside (as student because row 55 is inside of the `uo=moodle` (see lines: 19..31) `uo=student` (see lines: 33..45)) the course with name described in row 54 because enrol_ldap_course_idnumber is `cn`.

Once it has the list of the objects that fulfill all those conditions, it extracts the values of 'enrol_ldap_course_idnumber', 'enrol_ldap_course_shortname', 'enrol_ldap_course_fullname' and 'enrol_ldap_course_summary' attributes to use them as the course ID Number, short name, full name and summary (the first one to enrol users, the rest if you want the LDAP plugin to auto-create courses on the fly if they don't already exist).

Question05: Is the course setting: Enrolment Plugins relevant?
Answer05: No it isn't, as that setting is only relevant for interactive enrolment plugins (manual, paypal, autorize, etc.).

Question06: Is the course setting: Availability relevant?
Answer06: Yes it is. If a course is not available, Moodle won't enrol the user there (so it doesn't appear in the user course list). As soon as you make is available (visible) again, Moodle will start enrolling users into it

Question07: Is the course setting: Course enrollable relevant?
Answer07: No it isn't, as explained in Answer05.