
LDAP structures that can be used with Enhydra Shark's LDAP implementation of UserGroup and Authentication API

Table of Contents

Introduction	1
LDAP structure, type 0	1
LDAP structure, type 1	3

Introduction

The Lightweight Directory Access Protocol (LDAP) is a lightweight version of the Directory Access Protocol, which is part of X.500. Being neither a directory nor a database, LDAP is an access protocol that defines operations for how clients can access and update data in a directory environment.

At the moment, shark's LDAP implementation of UserGroup and Authentication API supports two types of LDAP structures. The first structure is marked as type 0, and the second is marked as type 1.

LDAP structure, type 0

This is simple LDAP structure. It contains groups and users. The list of LDAP object classes representing group of users is defined by configuration parameter *LDAPGroupObjectClasses*. If not defined, the default value is:

```
LDAPGroupObjectClasses=organizationalUnit
```

The list of LDAP object classes representing user is defined by configuration parameter *LDAPUserObjectClasses*. If not defined, the default value is:

```
LDAPUserObjectClasses=inetOrgPerson
```

Neither groups, nor users have an attribute that contains information saying to which group(s) the user (or group) belongs to. So, one user (or group) can belong to only one group. The only belonging of one user (or group) to only one group is defined by its dn (distinguished name). The *LDAPGroupUniqueAttributeName* parameter defines the name of attribute that is mandatory for each LDAP object class representing group of users. The value of this attribute MUST be unique for each LDAP entry for these object classes through the LDAP tree. If not defined, the default value is:

```
LDAPGroupUniqueAttributeName=ou
```

The *LDAPUserUniqueAttributeName* parameter defines the name of attribute that is mandatory for each LDAP object class representing user. The value of this attribute MUST be unique for each LDAP entry for these object classes throughout the LDAP tree. If not defined, the default value is:

```
LDAPUserUniqueAttributeName=userid
```

For example, the following data can represent the structure type 0:

```
version: 1
dn: o=SunsetComputers, c=sr
objectClass: top
objectClass: organization
o: SunsetComputers
version: 1
```

```
dn: userid=sasaboy, ou=developers, ou=programers, o=SunsetComputers, c=sr
objectClass: top
objectClass: inetOrgPerson
cn: Sasa Smith
givenname: Sale
initials: S.S.
mail: sasasmith@sunsetcomputers.com
mobile: 067/66688844
postaladdress: Tm92aSBTYWQsIFNla3NwaXJvdmEgNS8xMDAJ
postofficebox: 21000
sn: Smith
st: Serbia
street: 6th street 74
title: B.S.C. in E.E.
userid: sasaboy
userpassword:: c2FzYWJveQ==
```

```
dn: userid=simbe, ou=designers, ou=programers, o=SunsetComputers, c=sr
objectClass: top
objectClass: inetOrgPerson
cn: Sean Young
givenname: Sean
initials: S.Y.
mail: seanyoung@sunsetcomputers.com
mobile: 067/88833366
postaladdress: Tm92aSBTYWQsIFNla3NwaXJvdmEgNS8xMDAJ
postofficebox: 21000
sn: Young
st: Serbia
street: 4th street 27
title: B.S.C. in E.E.
userid: simbe
userpassword:: c2ltYmU=
```

```
dn: ou=programers, o=SunsetComputers, c=sr
objectClass: top
objectClass: organizationalUnit
ou: programers
```

```
dn: ou=developers, ou=programers, o=SunsetComputers, c=sr
objectClass: top
objectClass: organizationalUnit
ou: developers
```

```
dn: ou=designers, ou=programers, o=SunsetComputers, c=sr
objectClass: top
objectClass: organizationalUnit
ou: designers
```

In this example, there are three groups:

- programers

- developers
- designers

and two users:

- sasaboy
- simbe

The group *developers* belongs to group *programers*. It is defined by its dn: ou=developers, **ou=programers**, o=SunsetComputers, c=sr. The group *designers* also belongs to group *programers* (its dn: ou=designers, **ou=programers**, o=SunsetComputers, c=sr).

The user *sasaboy* belongs to group *developers* (its dn: userid=sasaboy, **ou=developers**, ou=programers, o=SunsetComputers, c=sr). The user *simbe* belongs to group *designers* (its dn: userid=simbe, **ou=designers**, ou=programers, o=SunsetComputers, c=sr).

LDAP structure, type 1

This is more complex LDAP structure. It also contains groups and users. The parameters *LDAPGroupObjectClasses*, *LDAPUserObjectClasses*, *LDAPGroupUniqueAttributeName* and *LDAPUserUniqueAttributeName* are used in the same way as in the structure type 0. Beside users and groups, in this structure type, is provided possibility of defining relations ("belong to") between groups and groups and between groups and users. The list of LDAP object classes representing relations between shark users and group or between shark groups is defined by configuration parameter *LDAPRelationObjectClasses*. If not defined, the default value is:

```
LDAPRelationObjectClasses=groupOfNames
```

The two attributes are important for these object classes. The *LDAPRelationUniqueAttributeName* parameter defines the name of attribute that is mandatory for each LDAP object class representing relation. The value of this attribute MUST be unique for each LDAP entry for these object classes throught the LDAP tree. If not defined, the default value is:

```
LDAPRelationUniqueAttributeName=cn
```

The *LDAPRelationMemberAttributeName* parameter defines the name of attribute of LDAP object classes (representing relation) that represents member that is included (user or group) in the relation - member is user or group that belongs to the group that is also defined in the relation. If not defined, the default value is:

```
LDAPRelationMemberAttributeName=member
```

.

For example, the following data can represent the structure type 1:

```
version: 1
dn: o=SunsetComputers, c=sr
objectClass: top
objectClass: organization
o: SunsetComputers
version: 1

dn: ou=Groups, o=SunsetComputers, c=sr
objectClass: top
objectClass: organizationalUnit
ou: Groups
```

dn: ou=Users, o=SunsetComputers, c=sr
objectClass: top
objectClass: organizationalUnit
ou: Users

dn: ou=GroupRelations, o=SunsetComputers, c=sr
objectClass: top
objectClass: organizationalUnit
ou: GroupRelations

dn: ou=UserRelations, o=SunsetComputers, c=sr
objectClass: top
objectClass: organizationalUnit
ou: UserRelations

dn: ou=programers, ou=Groups, o=SunsetComputers, c=sr
objectClass: top
objectClass: organizationalUnit
ou: programers

dn: ou=designers, ou=Groups, o=SunsetComputers, c=sr
objectClass: top
objectClass: organizationalUnit
ou: designers

dn: ou=developers, ou=Groups, o=SunsetComputers, c=sr
objectClass: top
objectClass: organizationalUnit
ou: developers

dn: ou=testers, ou=Groups, o=SunsetComputers, c=sr
objectClass: top
objectClass: organizationalUnit
ou: codeDesigners

dn: userid=sasaboy, ou=Users, o=SunsetComputers, c=sr
objectClass: top
objectClass: inetOrgPerson
cn: Sasa Smith
givenname: Sale
initials: S.S.
mail: sasasmith@sunsetcomputers.com
mobile: 067/66688844
postaladdress: Tm92aSBTYWQsIFNla3NwaXJvdmEgNS8xMDAJ
postofficebox: 21000
sn: Smith
st: Serbia
street: 6th street 74
title: B.S.C. in E.E.
userid: sasaboy
userpassword:: c2FzYWJveQ==

dn: userid=simbe, ou=Users, o=SunsetComputers, c=sr
objectClass: top
objectClass: inetOrgPerson
cn: Sean Young
givenname: Sean
initials: S.Y.
mail: seanyoung@sunsetcomputers.com
mobile: 067/88833366
postaladdress: Tm92aSBTYWQsIFNla3NwaXJvdmEgNS8xMDAJ
postofficebox: 21000
sn: Young

```
st: Serbia
street: 4th street 27
title: B.S.C. in E.E.
userid: simbe
userpassword:: c2ltYmU=
```

```
dn: cn=testers, ou=UserRelations, o=SunsetComputers, c=sr
objectClass: top
objectClass: groupOfNames
cn: testers
member: userid=sasaboy, ou=Users, o=SunsetComputers, c=sr
```

```
dn: cn=developers, ou=UserRelations, o=SunsetComputers, c=sr
objectClass: top
objectClass: groupOfNames
cn: developers
member: userid=simbe, ou=Users, o=SunsetComputers, c=sr
```

```
dn: cn=SunsetComputers, ou=GroupRelations, o=SunsetComputers, c=sr
objectClass: top
objectClass: groupOfNames
cn: SunsetComputers
member: ou=programers, ou=Groups, o=SunsetComputers, c=sr
```

```
dn: cn=programers, ou=GroupRelations, o=SunsetComputers, c=sr
objectClass: top
objectClass: groupOfNames
cn: programers
member: ou=designers, ou=Groups, o=SunsetComputers, c=sr
member: ou=developers, ou=Groups, o=SunsetComputers, c=sr
```

```
dn: cn=designers, ou=GroupRelations, o=SunsetComputers, c=sr
objectClass: top
objectClass: groupOfNames
cn: designers
member: ou=testers, ou=Groups, o=SunsetComputers, c=sr
```

```
dn: cn=developers, ou=GroupRelations, o=SunsetComputers, c=sr
objectClass: top
objectClass: groupOfNames
cn: developers
member: ou=testers, ou=Groups, o=SunsetComputers, c=sr
```

In this structure, four artificial groups must be created. The first is group that contains all groups. Its name is defined by parameter *LDAPGroupGroupsName*. If not defined, the default value is:

```
LDAPGroupGroupsName=Groups
```

In the example, this group is defined as:

```
dn: ou=Groups, o=SunsetComputers, c=sr
objectClass: top
objectClass: organizationalUnit
ou: Groups
```

The second is group that contains all users. Its name is defined by parameter *LDAPGroupUsersName*. If not defined, the default value is:

```
LDAPGroupUsersName=Users
```

In the example, this group is defined as:

```
dn: ou=Users, o=SunsetComputers, c=sr
objectClass: top
objectClass: organizationalUnit
ou: Users
```

The third is group that contains all relations between groups. Its name is defined by parameter *LDAPGroupGroupRelationsName*. If not defined, the default value is:

```
LDAPGroupGroupRelationsName=GroupRelations
```

In the example, this group is defined as:

```
dn: ou=GroupRelations, o=SunsetComputers, c=sr
objectClass: top
objectClass: organizationalUnit
ou: GroupRelations
```

The fourth is group that contains all relations between groups and users. Its name is defined by parameter *LDAPGroupUserRelationsName*. If not defined, the default value is:

```
LDAPGroupUserRelationsName=UserRelations
```

In the example, this group is defined as:

```
dn: ou=UserRelations, o=SunsetComputers, c=sr
objectClass: top
objectClass: organizationalUnit
ou: UserRelations
```

In this example, four groups are defined (they all belong to the group *Groups* - look their dn):

- programmers (dn is ou=programers, **ou=Groups**, o=SunsetComputers, c=sr)
- developers (dn is ou=developers, **ou=Groups**, o=SunsetComputers, c=sr)
- designers (dn is ou=designers, **ou=Groups**, o=SunsetComputers, c=sr)
- testers (dn is ou=testers, **ou=Groups**, o=SunsetComputers, c=sr)

and two users (they belong to the group *Users* - look their dn):

- sasaboy (dn is userid=sasaboy, **ou=Users**, o=SunsetComputers, c=sr)
- simbe (dn is userid=simbe, **ou=Users**, o=SunsetComputers, c=sr)

The groups *developers* and *designers* belong to group *programers*. In the example, this is defined as:

```
dn: cn=programers, ou=GroupRelations, o=SunsetComputers, c=sr
objectClass: top
objectClass: groupOfNames
cn: programers
member: ou=designers, ou=Groups, o=SunsetComputers, c=sr
member: ou=developers, ou=Groups, o=SunsetComputers, c=sr
```

Note that in the object class representing GroupRelations (in the example that is *groupOfNames*) has the value of unique relation attribute (in the example that is **cn**) set to the name of the group that contains the groups whose dn-s are defined in **member** attributes. This is the convention used in the structure type 1.

The group *testers* belongs to groups *developers* and *designers*. In the example, this is defined as:

```
dn: cn=designers, ou=GroupRelations, o=SunsetComputers, c=sr
objectClass: top
objectClass: groupOfNames
cn: designers
member: ou=testers, ou=Groups, o=SunsetComputers, c=sr
```

```
dn: cn=developers, ou=GroupRelations, o=SunsetComputers, c=sr
objectClass: top
objectClass: groupOfNames
cn: developers
member: ou=testers, ou=Groups, o=SunsetComputers, c=sr
```

So, in this structure type, a group can belong to more than group.

The user *sasaboy* belongs to group *testers* and the user *simbe* belongs to group *developers*. In the example, this is defined as:

```
dn: cn=testers, ou=UserRelations, o=SunsetComputers, c=sr
objectClass: top
objectClass: groupOfNames
cn: testers
member: userid=sasaboy, ou=Users, o=SunsetComputers, c=sr
```

```
dn: cn=developers, ou=UserRelations, o=SunsetComputers, c=sr
objectClass: top
objectClass: groupOfNames
cn: developers
member: userid=simbe, ou=Users, o=SunsetComputers, c=sr
```

The same as in the GroupRelations, the object class representing UserRelations (in the example that is *groupOfNames*) has the value of unique relation attribute (in the example that is **cn**) set to the name of the group that contains the users whose dn-s are defined in **member** attributes. This is the convention used in the structure type 1.

The same way as a group can belong to more than one group, and user can belong to more than one group.