



## LDAP-Driven Accounts Using Group Membership

Authentication and assigning users to User Classes can be filter down to their group membership level. This offers greater flexibility with filtering users when they may exist in the same Organization Unit or Container and allows you to grant users to multiple Class Definitions and their assigned payment methods.



**Note:** the following steps pertain to managing both end-users, as well as users who can be assigned to Administrative Classes and granted various levels of system administration.

### Requirements

An existing LDAP container must exist and successfully binding. For more information see the section on creating LDAP Authentication Connectors.

For configuration options please refer to the following guides:

[http://www.goprintsupport.com/Active\\_Directory\\_LDAP.pdf](http://www.goprintsupport.com/Active_Directory_LDAP.pdf)

[http://www.goprintsupport.com/Scheduled\\_Quota\\_LDAP\\_Users.pdf](http://www.goprintsupport.com/Scheduled_Quota_LDAP_Users.pdf)

Accounts – Authentication Connectors:

Name	Type	Active
students	LDAP	Active



**Sample:** LDAP Connector

**LDAP Server**

Enter the LDAP server's network name.

Server Name	<input type="text" value="gs4srv"/>
Security	<input type="text" value="Simple (no network privacy)"/>

**Search Target**

Set the Base DN that is the top-level target for searches.

Base DN	<input type="text" value="DC=goprintcorp,DC=dyndns,DC=org"/>
---------	--

**Search User Account**

Enter the DN of a user account to use when searching the LDAP repository. Most LDAPs require an authenticated user to perform searching, and one is recommended even if left blank, an 'anonymous bind' will be used to search.

Search User DN	<input type="text" value="CN=ldapuser,CN=Users"/>
Password	<input type="text" value=""/> <a href="#">Previously Set</a>

Step 1 – Select **NONE** at the LDAP Connector Attribute section

From the Default Class drop down menu select **NONE**

- ✔ **Important:** Setting the Default Class level to None forces the LDAP search to first authenticate Users then if a group membership exists at the Class Definition level, then users are granted access to the payment method.

Account ID	<input type="text" value="sAMAccountName"/>
Default Class	<input type="text" value="None"/>
Class Name	<input type="text"/>
Card Number	<input type="text"/>
	<input type="checkbox"/> <b>Lookup via database query</b>
First Name	<input type="text" value="givenName"/>
Last Name	<input type="text" value="sn"/>
Reference Number	<input type="text"/>
E-Mail	<input type="text" value="mail"/>
	<a href="#">Attribute Browser</a>



## Step 2 – Select LDAP Options

Navigate to Accounts – Class Definitions

Select the desired User Class and select LDAP Options

**Properties** | **LDAP Options**

**Class Type** USER  
The USER class is used for users. Users can be members of one or more classes. Users can be members of one or more purses. Purses are the unique combination of Class and Pay Method, and represent accounts at the transaction level for proper accounting treatments.

**Class Name** Default User Class

**Pay Methods**  
Mark the Pay Methods that you want to allow for this class and

Type	Purse Name
<input checked="" type="checkbox"/> Quota	free print funds
<input type="checkbox"/> Allowance	Allowance

## Step 3 – Enter the corresponding group membership syntax

### Option 1 - Group membership Accounts Using Distinguished Names

Every entry in the directory has a distinguished name (DN). The DN is the name that uniquely identifies an entry in the directory. A DN is made up of attribute=value pairs, separated by commas. This is the easiest way to drive Class membership based on data in the LDAP

Simply provide the full DN of the group container that is associated with this Class of users.

**Example:** When it's not necessary specify a complex memberOf string; you can use the built-in distinguished name of the group. Note: Nested OU's are supported.

CN=it,OU=staff,DC=goprint,DC=edu
CN=Students,OU=Campus,OU=Groups,OU=Managed,DC=goprint,DC=edu



## Option 2 –Group Membership LDAP String using MemberOf Attribute



**Note:** Each argument must exist in its own set of parentheses. The entire LDAP statement must be encompassed in a main set of parentheses.

Scenario #1 – Single group membership

*(MemberOf=CN=students,DC=goprintcorp,DC=dyndns,DC=org)*

### Method 2: LDAP Filter

Membership in this Class can alternatively be driven by LDAP attributes.

Build an LDAP filter below and this Class will be automatically assigned to use match the filter.

The LDAP Filter specified here is executed in the context of the authenticated users present in the user's LDAP container and readable by the user are accessible.

```
(memberOf=CN=students,DC=goprintcorp,DC=dyndns,DC=org)
```



## Scenario #2 – Matching Multiple Groups

**& (logical AND)** - More than one condition, and you want all conditions in the series to be true.

```
(|(memberOf=CN=medstudents,DC=goprintcorp,DC=dyndns,DC=org)(memberOf=CN=lawstudents,DC=goprintcorp,DC=dyndns,DC=org))
```

The & operator states that all Arguments must be true, or match. In this case, the matching users MUST be a member of **BOTH** groups, ITS and staff.

```
(&(memberOf=CN=ITS,DC=goprintcorp,DC=dyndns,DC=org)
(memberOf=CN=staff,DC=goprintcorp,DC=dyndns,DC=org))
```

The & operator states that all Arguments must be true, or match. In this case, the matching users MUST be a member of BOTH groups, ITS and staff.

## Scenario #3 – Matching Multiple Groups

**| (logical or)** – either condition is true

```
(|(memberOf=CN=med students,DC=goprintcorp,DC=dyndns,DC=org)(memberOf=CN=lawstudents,DC=goprintcorp,DC=dyndns,DC=org))
```

The | Operator states that EITHER Argument can be true. In this case, users can be a member of either group med students or law students.

```
(|(memberOf=CN=med students,DC=goprintcorp,DC=dyndns,DC=org)
(memberOf=CN=law students,DC=goprintcorp,DC=dyndns,DC=org))
```

The | operator states that either Argument can be true. In this case, users can be a member of either group, med students and law students



#### Scenario #4 – Excluding Multiple Groups

**! (logical NOT)** - exclude objects that have a certain attribute

```
(!(memberOf=CN=med students,DC=goprintcorp,DC=dyndns,DC=org)(memberOf=CN=law students,DC=goprintcorp,DC=dyndns,DC=org))
```

The ! Operator states that the first Argument must be true and NOT the second. In this case, the Argument MUST match the users in the group med students, and exclude users in the group students.

A screenshot of a text box with a scroll bar on the right. The top part contains the LDAP filter code: 

```
(!(memberOf=CN=med students,DC=goprintcorp,DC=dyndns,DC=org)(memberOf=CN=students,DC=goprintcorp,DC=dyndns,DC=org))
```

. Below the code, there is a red-colored explanatory text: 

The ! operator states that the first Argument must be true and NOT the second. In this case, the Argument MUST match the users in the group med students, and exclude users in the group students.