

# LDAP / SSO Authentication

Extension Key: ig\_ldap\_sso\_auth

Language: en

Keywords: ldap, sso, authentication

Copyright 2000-2010, Michael Miousse, <mmiousse@infoglobe.ca>

This document is published under the Open Content License  
available from <http://www.opencontent.org/opl.shtml>

The content of this document is related to TYPO3  
- a GNU/GPL CMS/Framework available from [www.typo3.org](http://www.typo3.org)

## Table of Contents

LDAP / SSO Authentication.....	1	Screenshots.....	3
<b>Introduction</b> .....	<b>3</b>	<b>Users manual</b> .....	<b>7</b>
What does it do?.....	3	<b>Administration</b> .....	<b>8</b>

# Introduction

## What does it do?

- This extension enables import/update/deletion of users and groups(frontend, backend or both) from a LDAP-directory. In this way, TYPO3 can be used as an intranet CMS. Multiple ldap sever configuration are allowed. Works with openldap, edirectory and Active directory. You can also use a CAS server to implement SSO.

## Screenshots

Path: New TYF

Edit Configuration LDAP / SSO "local" on root level

GENERAL LDAP BE\_USERS BE\_GROUPS FE\_USERS FE\_GROUPS CAS

Server : OpenLDAP

Protocol : 3

Charset : utf-8

Host : localhost

Port : 389

Bind DN : cn=admin,dc=example,dc=com

Password : \*\*\*\*\*

Configuration LDAP / SSO [1]

Edit Configuration LDAP / SSO "local" on root level

GENERAL LDAP BE\_USERS BE\_GROUPS FE\_USERS FE\_GROUPS CAS

Base DN : ou=Users,dc=example,dc=com

Filter : (uid={USERNAME})

Mapping :

```
tstamp = {DATE}
realName = <cn>
email = <email>
lang = fr
```

### Edit Configuration LDAP / SSO "local" on root level

GENERAL
LDAP
BE\_USERS
BE\_GROUPS
FE\_USERS
FE\_GROUPS
CAS

**Base DN :**

ou=Groups,dc=example,dc=com

**Filter :**

&(objectClass=posixGroup)

**Mapping :**

tstamp = {DATE}

TYP03

WEB

Page
View
List
Info
Access
Functions
Template

MEDIA

File
List
Info
Tools

USER TOOLS

Task center
User settings

ADMIN TOOLS

LDAP / SSO

User Admin
Ext Manager
DB check
Configuration
Install
Log
Scheduler
phpLDAPadmin

HELP

About Modules
TypoScript Help

LDAP / SSO service configuration

Status

View configuration local (1)

LDAP

server	OpenLDAP
host	localhost
port	389
protocol	3
charset	utf-8
binddn	cn=admin,dc=example,dc=com
password	*****

Connexion status

connect	<table> <tr><td>host</td><td>localhost</td></tr> <tr><td>port</td><td>389</td></tr> <tr><td>status</td><td>Success</td></tr> </table>	host	localhost	port	389	status	Success
host	localhost						
port	389						
status	Success						
bind	<table> <tr><td>dn</td><td>cn=admin,dc=example,dc=com</td></tr> <tr><td>password</td><td>*****</td></tr> <tr><td>status</td><td>Success</td></tr> </table>	dn	cn=admin,dc=example,dc=com	password	*****	status	Success
dn	cn=admin,dc=example,dc=com						
password	*****						
status	Success						

CAS

CAS authentication disable.

Backend authentication

LDAPAuthentication	1
CASAuthentication	0
DeleteCookieLogout	0
forceLowerCaseUsername	1
evaluateGroupsFromMembership	0
IfUserExist	0
IfGroupExist	0
BEfailsafe	1
DeleteUserIfNoLDAPGroups	0
DeleteUserIfNoTYPO3Groups	0
GroupsNotSynchronize	1
requiredLDAPGroups	0
updateAdminAttribForGroups	0

TYP03

▼ WEB

Page

View

List

Info

Access

Functions

Template

▼ MEDIA

File

List

Info

Tools

▼ USER TOOLS

Task center

User settings

▼ ADMIN TOOLS

LDAP / SSO

User Admin

Ext Manager

DB check

Configuration

Install

Log

Scheduler

phpLDAPadmin

▼ HELP

About Modules

TypoScript Help

LDAP / SSO service configuration

Search wizard ▼

View configuration local (1)

Search wizard

be\_users

fe\_users

be\_groups

fe\_groups

First entry only

See status

Base DN : ou=Users,dc=example,dc=com


Filter : (uid=\*)

search

Result

uid	count 1	0 mmousse
0	uid	
cn	count 1	0 Michael
1	cn	
sn	count 1	0 Mousse
2	sn	
userPassword	count 1	0 12345
3	userPassword	
loginShell	count 1	0 /bin/bash
4	loginShell	
uidNumber	count 1	0 100
5	uidNumber	
gidNumber	count 1	0 100
6	gidNumber	
homeDirectory	count 1	


Information

**The extension is installed (loaded and running)!**  
Click here to remove the extension: 

**Configuration:**

(Notice: You may need to clear the cache after the configuration of the extension. This is required if the extension adds TypoScript depending on these

**Configuration check** [checkConfiguration]

 **No errors were found**  
Configuration has been configured correctly.

**Throw Exception if misconfigured** [throwExceptionAtLogin]  
Recommended for new installation - LDAP authentication is not compatible with loginSecurityLevel set to "challenged" or "superchallenged" since the real password can never be sent against the LDAP repository. Instead of failing silently, the extension will throw an Exception when login. If the method appear to be brutal it will save a log of headaches. Value of loginSecurityLevel should be handled manually to "normal" or even better "rsa" in the Install Tool.

☒

**Use configuration** [uidConfiguration]  
Uid of configuration record in table tx\_igldapssoauth\_config.

**Force lowercase username** [forceLowerCaseUsername]  
If checked, usernames will be always be stored lowercased. Usefull for case-sensitive databases.

☒

**Enable features**

**Groups from membership** [evaluateGroupsFromMembership]  
If checked, obtain groups from user with membership attribute. Else, obtain groups object indirectly with DN of user.

☐

**Backend LDAP authentication** [enableBELDAPAuthentication]  
Enable LDAP authentication for the backend.

☒

**If backend user exist locally** [TYPO3BEUserExist]  
If checked, user not found in the table be\_users may not log on.

☐

**backend user failsafe** [BEfailsafe]  
If checked, backend users found in the database can log on only if the username does not exist in the LDAP.

☒

**Not synchronize the BE groups** [TYPO3BEGroupsNotSynchronize]

# Users manual

When a user tries to log on the extension passes the credentials to the LDAP server(s) and verifies them. When a LDAP server can authenticate the user he is logged on. New users (in the directory but not in the TYPO3 database) are imported after authentication. Records of existing users are updated.

For new users imported from the directory random passwords will be inserted!

# Administration

-First step for configuring your LDAP authentication is to create one or more server configurations. All server configurations have to be stored on the root level of your TYPO3 website. Create a Configuration LDAP / SSO records. These records have 7 tabs for specific configurations types.

1. The first tab is General.
  - The only thing you have to fill is the configuration name. This is only to name the records you have just created.
2. The second tab is the Global configurations about a single LDAP (not that you can create multiple configuration records with the same LDAP)
  - **Server** : Choose your LDAP type (openldap or edirectory).  
Note that if you are using Active directory, your LDAP type is edirectory
  - **Protocol** : Choose the LDAP protocol version (2 or 3). Recent LDAP use version 3
  - **Charset** : is Charset of your LDAP. Usually utf-8
  - **Host** : is the host of your LDAP
  - **Port** : is the port your LDAP use. Default LDAP port is 389
  - **Bind DN** : is the full DN of the LDAP user you will use to connect to the LDAP  
note that your LDAP user need access to the directory where users and groups are stored and full read access to users and groups.  
Example: cn=admin,dc=example,dc=com
  - **Password** : Password of the user used to connect to the LDAP
3. The third tab and the fifth tabs can be fill exactly the same way. The only difference between them is that BE\_USERS store the configurations for the backend LDAP user authentication and FE\_USERS store the configurations for the frontend LDAP user authenticate. You will only fill the section you need, BE\_USERS if you need backend authentication and FE\_USERS if you need frontend authentication.
  - **Base DN** : is the full DN path of the directory containing all the users that you want to use with you TYPO3 authentication.  
Example: ou=Users,dc=example,dc=com
  - **Filter** : is used to precise which LDAP attribute contain the username of your users.  
Example: (uid={USERNAME})  
uid is the most common attribute used to keep the username in LDAPs but if you are in an Active directory, the field where the username is stored is usually the sAMAccountName.  
Note that the string "{USERNAME}" will be substituted by the username entered in the login form.  
You will also be able to add restrictions that allow you to exclude user from specific properties. The syntax used in this field is the LDAP search syntax.  
Example: (&(uid={USERNAME})(objectClass=user))
  - **Mapping** : is used to fetch other attributes form the LDAP that we would like users to have. It is quite simple, each line is a new command. Each command have 2 parts separated by a =. the first part is the field form the TYPO3 user that we want to fill and the second part is the value we that the field will have. There is 3 possible value type you could use:
    - a string : this will assign the value directly to the field
    - a LDAP attribute value: LDAP attributes will be recognized by the specific characters <>.  
Example: email = <email> this will set the field email of the TYPO3 user to the value of the attributes email of the user fetch from the LDAP
    - a custom marker: custom markers are markers create by the extension to assign specific type of values. There are only 4 markers available for the moment:
      - {DATE}: the current timestamp



- {RAND}: a random number
- {USERNAME}: the username from the login form ( the username will automatically fill the needed field. This markers is only used if you want to put the username in an other field than the one by default )
- {hookparameters} : will only be usefull if an extension il hooked on ig\_ldap\_sso\_auth

Example:   tstamp = {DATE}  
               realName = <cn>  
               email = <email>  
               lang = fr

4. The fourth and sixth tabs can be fill exactly the same way. The only difference between them is that BE\_GROUPS store the configurations for the backend LDAP usergroup association and FE\_GROUPS store the configurations for the frontend LDAP usergroup association. You will only fill the sections you need, BE\_GROUPS if you need backend authentication and FE\_GROUPS if you need frontend authentication. You can skip this entire section if you just want to validate the authentication and do not want to use groups from LDAP.

- **Base DN** : is the full DN path of the directory containing all the groups that are related to your LDAP users and you want to use in your TYPO3.

Example: ou=Groups,dc=example,dc=com

- **Filter** : You will only by used to add restrictions that allow you to exclude ojects from specific properties. The syntax used in this field is the LDAP search syntax.

Example: &(objectClass=posixGroup)

- **Mapping** : is used to fetch other attributes form the LDAP that we would like groups to have. It is quite simple, each line is a new command. Each command have 2 parts separated by a =. the first part is the field form the TYPO3 group that we want to fill and the second part is the value we that the field will have. There is 3 possible value type you could use:

- a string : this will assign the value directly to the field
- a LDAP attribute value: LDAP attributes will be recognized by the specific caracters <>.

Example: email = <email> this will set the field email of the TYPO3 group to the value of the attributes email of the user fetch from the LDAP

- a custom marker: custom markers are markers create by the extension to assgin specific type of values. There are only 4 markers available for the moment:

- {DATE}: the current timestamp
- {RAND}: a random number
- {USERNAME}: the username from the login form ( the username will automatically fill the needed field. This markers is only used if you want to put the username in an other field than the one by default )
- {hookparameters} : will only be usefull if an extension il hooked on ig\_ldap\_sso\_auth

Example:   tstamp = {DATE}

5. The last tab is for CAS configurations. You only have to fill it if you want to use a CAS server to implement some single sing on (SSO)

- **Host** : is the host of your CAS server
- **URI** : path to postpend to the host used if the CAS sever is not at the root of your host

Example: /userSSo/cas in the string localhost/userSSo/cas

- **Service URL** : this is a specific url for your CAS
- **Port** : port on which you CAS is configure
- **Back URL** : Url to return to in case of a CAS login