EXT: LDAP Integration

Extension Key: ldap
Language: en
Keywords: LDAP, Active Directory, Authentication, forAdmins
Copyright 2000-2013, Norman Seibert, <seibert(at)entios.de>

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

Table of Contents

Introduction

What does it do?

This extension allows TYPO3 to connect to LDAP directories and to fetch user records from them. Features include:

- Handling of multiple LDAP servers
- Storage of LDAP server configurations in the TYPO3 database or a configuration file
- Import/Update/Delete of frontend (FE) and backend (BE) users
- Import of user groups
- Flexible mapping of LDAP attributes to TYPO3 user properties
- Authentication of FE and BE users against the directory
- Usage of the TYPO3 scheduler to import/update/delete TYPO3 users

Compatibility

- Version 3.x of eu_ldap is not compatible with LDAP server records created with 2.x, instead you have to redefine your server records in the configuration file.
- Version 3.1 supports TYPO3 6.0 and higher.

Users manual

Installation

1. Install the extension through the extension manager.
2. Set the extension's basic settings in the extension manager.
   For the frontend login it is necessary to specify a root page id due to some Extbase bugs in TYPO3 6.0.
3. Configure the authentication mode, whether you want to enable FE or BE users to login using their LDAP credentials. Please note that enabling BE authentication and setting exclusive authentication against LDAP may prevent you from logging into the TYPO3 backend! Test first!!!
4. The extension can log errors or single execution steps to the TYPO3 developer log. To use it, please install the extension “devlog”! If you set the logging level to “2” all activities – even user credentials – are logged for debugging purposes.
5. Create LDAP server records in your configuration file.
6. Use the LDAP backend module to check your configuration.

The backend module

The backend module provides functions to:

– Get an overview of your LDAP server records

– Import users

– Update users

– Delete users who are not in the directory

– Check login against LDAP

Using the extension from the command line

The extension provides a so called “Command Controller” which can be invoked via the command line:

```
./typo3/cli_dispatch.phpsh extbase ldap:<function> <parameters>
```

The following functions are supported:

Function	Description	Parameters
importUsers	Imports new users	**servers [string]** comma separated list (no spaces) of server uids from the configuration file **processFe [boolean, 0/1]** Import frontend users **ProcessBe [boolean, 0/1]** Import backend users
updateUsers	Updates existing users	**servers [string]** comma separated list (no spaces) of server uids from the configuration file **processFe [boolean, 0/1]** Update frontend users **processBe [boolean, 0/1]** Update backend users

Function	Description	Parameters
importAndUpdateUsers	Imports new users and updates existing ones	**servers [string]** comma separated list (no spaces) of server uids from the configuration file **processFe [boolean, 0/1]** Import frontend users **processBe [boolean, 0/1]** Import backend users
deleteUsers	Deletes or disables users not found in any LDAP directory	**processFe [boolean, 0/1]** Import frontend users **processBe [boolean, 0/1]** Import backend users **hideNotDelete [boolean, 0/1]** Disable users instead of deleting them **deleteNonLdapUsers [boolean, 0/1]** Delete/deactivate also users which have not been imported from a directory

Creating Scheduler tasks

Using an Extbase Command Controller as a Scheduler Task allows scheduled execution of an action. To create a scheduled execution simply add a new Scheduler task with class “Extbase CommandController Task (extbase)”. Choose the CommandController Command, save the task and reopen it to set parameters. These are the same as described in the section above.

FAQ

– Is there a limit on the number of user records which can be imported from a directory?

No, there isn't – at least not in the extension. Many LDAP servers are configured to retrieve only 1000 records per search, so please check your LDAP server if you get only 1000 entries.

– Can I import nested user groups from an LDAP directory?

No, this is (currently) not supported.

Configuration

Correct configuration of LDAP server records is crucial and most problems result from wrong configurations. A general advice is to set the logging level to “2” in the extension's configuration (in the extension manager) and install the “devlog” extension.

FAQ

– FE users are authenticated correctly and imported from the LDAP directory but they cannot login. What's wrong?

In most cases the users do not belong to a user group and TYPO3 prevents them from login without one. Either correct the mapping for user group retrieval from the directory or configure one or more “standard” user groups for all LDAP users.

Reference

The following table lists the properties of an LDAP server record. If you manage your server records in a configuration file you will recognize the property names immediately, in the backend the properties may have different (and localized) labels.

The configuration file

The configuration file uses a Typoscript like syntax, the root element to be used is “ldapServers”.

Mandatory properties are printed bold.

Property	Data type	Description	Default
title	string	Server name	
disable	boolean	Disable the server record	0
host	string	The server's ip address or DNS name	
port	int+	The server's port, mostly 389	
forceTLS	boolean	Encrypt the connection even if using port 389 which is used for unencrypted connections by default	0
version	int+	The server's LDAP version, currently “3” should work for most servers	
authenticate	string	FE: Authenticate FE users BE: Authenticate BE users both: Authenticate FE and BE users	
user	string	User (DN) with read access to the directory	
password	string	The user's password	
fe_users.	COA	You have to set either “fe_users” or “be_users”, otherwise nothing will happen ...	
fe_users.pid	int	Page ID for user storage	
fe_users.baseDN	string	The BaseDN for all LDAP searches	
fe_users.filter	string	The LDAP query for user retrieval, “<search>” will be replaced by the user's username.	
fe_users.autoImport	boolean	If set users will be imported/updated automatically after login.	0

Property	Data type	Description	Default

fe_users.mapping.	COA	Configures the TYPO3 user table fields, the basic syntax is:
		`<TYPO3 table field>.data = field:<LDAP attribute>`
		
		If an LDAP attribute is used multiple times (multivalue) the resulting array will be imploded into a comma-separated list. In this case you have to use “stdWrap.” for all wrapping functions.
		
		The LDAP attribute have to be written in lowercase!
		
		Example
		The following example updates the table field “name” with the value “displayname” of the user's LDAP record and wraps it with stars:
		`name {`
		`  data = field:displayname`
		`  wrap = * | *`
		`}`	

fe_users.usergroups.	COA	Without a usergroup FE users are unable to login to TYPO3.	
fe_users.usergroups.importGroups	boolean	Import usergroups from the LDAP directory.	0
fe_users.usergroups.restrictToGroups	List of strings	Only import groups if the name satisfies the given pattern(s). Please note, that all wraps and stdWrap. is applied before. Regular expression. **Example** The following example imports only users which belong to a group beginning with “typo3” (case insensitive): `restrictToGroups = /^typo3.*/i`	
fe_users.usergroups.addToGroups	List of int+	Add each imported/updated user to this TYPO3 user group(s). Comma-separated list of usergroup UIDs.	
fe_users.usergroups.reverseMapping	boolean	If your LDAP directory stores users as group attributes (OpenLDAP) set this value to 1.	0
fe_users.usergroups.preserveNonLdapGroups	Boolean	Preserve relations to usergroups which have not been imported from an LDAP server	
be_users.	COA	Same as “fe_users” but property “pid” dœs not exist because BE users are stored on the root page (zero)	

Example

The following example configures a Windows active Directory server for FE user import:

```
ldapServers {
  localhost {
    title = lokaler LDAP-Server
    host = 127.0.0.1
    port = 389
    forceTLS = 0    # 0/1, true/false
    version = 3
    authenticate = FE
    user = uid=admin,ou=system
    password = secret
    fe_users {
      pid = 20
```

```
        baseDN = ou=users
        filter = (&(mail=*)(objectClass=user)(objectCategory=person)
(sAMAccountName=<search>))
        autoImport = 1
        mapping {
          name {
            data = field:displayname
            wrap = * | *
          }
          username.data = field:samaccountname    #mandatory
          address.data = field:streetaddress
          zip.data = field:postalcode
          city.data = field:l
          country.data = field:countrycode
          fax.data = field:facsimiletelephonenumber
          mail.data = field:mail
          phone.data = field:telephonenumber
          www.data = field:wwwhomepage
          timestamp.data = field:logintime
        }
        usergroups {
          importGroups = 1
          mapping {
            field = DN
            field.data = field:memberof
            title.data = field:name
            title.stdWrap.noTrimWrap = || (LDAP)
          }
          restrictToGroups =
          addToGroups = 3
        }
      }
    }
}
```

ChangeLog

Version	Changes:
3.1.0	Initial extbase version of ldap extension.