

Postfix with Virtual users (MailDir format) using openldap with back-sql (MySQL) -beta

This document is owned by ceylonlinux.com

A verbose step by step guide

Commercial support is also available. See <http://www.eylonlinux.com> for more information.

Version 1.0

Author: M.B.G.Suranga De Silva <suranga [at] ceylonlinux [dot] com>

Last edited 02/08/2006

- The name of my company is ceylonlinux (ceylonlinux.com)
- I am trying to share my experience with you. I am 100% sure this works, but do it on your own risk
- I did this in Linux
- All mails are owned by one virtual user, no Unix accounts needed.
- Everything is configured from ldap
- A loosely coupled system
- Mail is in maildir format and can be placed on NFS (The location of Maildir is /usr/local/vmail)
- All my source codes are in /usr/local/src
- Don't worry about the security for the moment keep on reading this doc. My next howto will address all the security settings in a hardened SELinux BOX.

System overview

- Postfix will act as MTA
- Openldap will give username, passwords, location of Mail directories and other related information
- MySQL will work as back-sql to the ldap (Here we are using mysql instead of bdb because we are having large number of existing users in our database)
- Dovecot will work as POP3 and IMAP servers

Installation

I downloaded postfix-2.2.10.tar.gz, openldap-2.3.24.tgz, libiodbc-3.52.4.tar.gz

extract libiodbc-3.52.4.tar.gz and install

```
[root@suranga src]#tar -xzvf libiodbc-3.52.4.tar.gz
[root@suranga src]#cd libiodbc-3.52.4
[root@suranga libiodbc-3.52.4]#./configure
[root@suranga libiodbc-3.52.4]#make
[root@suranga libiodbc-3.52.4]#make install
[root@suranga libiodbc-3.52.4]#cd ..
[root@suranga src]#tar -xzvf openldap-2.3.24.tar.gz
[root@suranga src]#cd openldap-2.3.24
[root@suranga openldap-2.3.24]#export ODBCINI=/etc/odbc.ini
[root@suranga openldap-2.3.24]#./configure --enable-sql=yes
[root@suranga openldap-2.3.24]#make depend
[root@suranga openldap-2.3.24]#make
[root@suranga openldap-2.3.24]#cd tests
[root@suranga tests]#SLAPD_USE_SQL=mysql
[root@suranga tests]#SLAPD_USE_SQLWRITE=yes
[root@suranga tests]#make sql
[root@suranga tests]#cd ..
[root@suranga tests]#make install
```

Here is my slapd.conf in /usr/local/etc/openldap

```
[root@suranga suranga]# more /usr/local/etc/openldap/slapd.conf
#
# See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.
#
include      /usr/local/etc/openldap/schema/core.schema
include      /usr/local/etc/openldap/schema/cosine.schema
include      /usr/local/etc/openldap/schema/inetorgperson.schema
include      /usr/local/etc/openldap/schema/nis.schema
include      /usr/local/etc/openldap/schema/corba.schema
include      /usr/local/etc/openldap/schema/java.schema
#This is our own schema
```

```

include      /usr/local/etc/openldap/schema/user.schema

# Define global ACLs to disable default read access.
access to *
    by * write

# Do not enable referrals until AFTER you have a working directory
# service AND an understanding of referrals.
#referral    ldap://root.openldap.org

allow bind_v2
pidfile      /usr/local/var/run/slapd.pid
argsfile     /usr/local/var/run/slapd.args

schemacheck  on

#####
# sql database definitions
#####

database     sql
suffix       "dc=ceylonlinux,dc=com"
rootdn       "cn=root,dc=ceylonlinux,dc=com"
rootpw       secret
dbname       ldap
dbuser       ldap
dbpasswd     password
subtree_cond "ldap_entries.dn LIKE CONCAT('%',?)"
insentry_query "INSERT INTO ldap_entries (dn,oc_map_id,parent,keyval) VALUES (?, ?, ?, ?)"
has_ldapinfo_dn_ru    no

```

[root@suranga libiodbc-3.52.4]# mysqladmin create ldap
Next we will create a MySQL account that OpenLDAP will use for our newly created ldap database

[root@suranga libiodbc-3.52.4]# mysql

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 10 to server version: 4.0.18

Type 'help;' or 'h' for help. Type 'c' to clear the buffer.

```
mysql>grant all privileges on ldap.* to 'ldap'@'localhost'  
->identified by 'password' with grant option;
```

Query OK, 0 rows affected (0.13 sec)

We of course want to substitute 'password' with the actual password we wish to use for this particular user account

Configuring LibODBC to use the MyODBC driver

Quite simply we need to edit two file here to get LibODBC to use the MyODBC driver in accessing the MySQL server.

Take a look at the odbcinst.ini and odbc.ini files and make the following changes

Here is the output of my odbc.ini and odbcinst.ini files

```
[root@suranga openldap-2.3.24]# more /etc/odbc.ini
```

```
[ODBC Data Sources]
```

```
ldap = MySQL LDAP DSN
```

```
[ldap]
```

```
Driver = /usr/lib/libmyodbc3.so
```

```
Description = OpenLDAP Database
```

```
Host = localhost
```

```
ServerType = MySQL
```

```
Port = 3306
```

```
FetchBufferSize = 99
```

```
User = ldap
```

```
Password = password
```

```
Database = ldap
```

```
ReadOnly = no
```

```
Socket = /var/lib/mysql/mysql.sock
```

```
[ODBC]
```

InstallDir=/usr/local/lib

```
[root@suranga /]# more /etc/odbcinst.ini
[MySQL ODBC 3.51 Driver]
DRIVER      = /usr/lib/libmyodbc3.so
FileUsage   = 1
```

```
[root@suranga /]#cd /usr/local/src/openldap-2.3.24/servers/slapd/back-sql/rdbms_depend/mysql/
```

```
[root@suranga mysql]#mysql < backsql_create.sql ldap -p
[root@suranga mysql]#mysql < testdb_create_ceylonlinux.sql ldap -p
[root@suranga mysql]#mysql < testdb_data_ceylonlinux.sql ldap -p
[root@suranga mysql]#mysql < testdb_metadata_ceylonlinux.sql ldap -p
```

Please refer above files carefully. I have generously added those in the appendix. Hope you will give me some credit for that. All you need is simply copy and paste :)

My schema (user.schema) I have developed for our university is also in the appendix. Dear Lecturers, hope you will also give me some credit on this.

Copy the schema file to the directory /usr/local/etc/openldap/schema/ as follows

```
[root@suranga suranga]#cp /usr/local/src/user.schema /usr/local/etc/openldap/schema/user.schema
```

```
[root@suranga openldap-2.3.20]# iodbctest
iODBC Demonstration program
This program shows an interactive SQL processor
Driver Manager: 03.52.0406.0126
```

Enter ODBC connect string (? shows list):

```
DSN                | Driver
-----
ldap                | MySQL LDAP DSN
```

Enter ODBC connect string (? shows list): DSN=ldap
Driver: 03.51.06 (libmyodbc3.so)

SQL>show tables;

Tables_in_ldap

authors_docs
documents
institutes
ldap_attr_mappings
ldap_entries
ldap_entry_objclasses
ldap_oc_mappings
persons
phones
referrals

result set 1 returned 10 rows.

SQL>exit

Have a nice day.[root@suranga openldap-2.3.20]#

[root@suranga openldap-2.3.20]#

[root@suranga openldap-2.3.20]# export ODBCINI=/etc/odbc.ini

[root@suranga openldap-2.3.20]#export LD_LIBRARY_PATH=/usr/local/lib/

[root@suranga libiodbc-3.52.4]# /usr/local/libexec/slapd -d 1

you can stop the debug mode by killing the relevant process number

Now run in daemon mode

[root@suranga libiodbc-3.52.4]# /usr/local/libexec/slapd

varify as follows

```
[root@suranga libiodbc-3.52.4]# ps -ef | grep slapd
root  31032 4053  0 00:20 pts/3  00:00:00 /usr/local/libexec/slapd
root  31072 4053  0 00:23 pts/3  00:00:00 grep slapd
```

```
[root@suranga libiodbc-3.52.4]# ldapsearch -x -b "dc=ceylonlinux,dc=com" "(objectclass=*)"
```

```
# extended LDIF
```

```
#
```

```
# LDAPv3
```

```
# base <dc=ceylonlinux,dc=com> with scope subtree
```

```
# filter: (objectclass=*)
```

```
# requesting: ALL
```

```
#
```

```
# Shantha Fernando, ceylonlinux.com
```

```
dn: cn=Shantha Fernando,dc=ceylonlinux,dc=com
```

```
objectClass: user
```

```
cn: Shantha Feranado
```

```
sn: Feranado
```

```
mail: shantha@ceylonlinux.com
```

```
seeAlso: documentTitle=book1,dc=ceylonlinux,dc=com
```

```
seeAlso: documentTitle=book2,dc=ceylonlinux,dc=com
```

```
givenName: Shantha
```

userUserID: shantha

userFaculty: cse

userPassword:: bWl0

telephoneNumber: 222-3234

telephoneNumber: 332-2334

mailMessageStore: /usr/local/vmail/ceylonlinux.com/shantha/

Gihan Dias, ceylonlinux.com

dn: cn=Gihan Dias,dc=ceylonlinux,dc=com

objectClass: user

cn: Gihan Dias

sn: Dias

seeAlso: documentTitle=book1,dc=ceylonlinux,dc=com

givenName: Gihan

telephoneNumber: 545-4563

Vishaka Nanayakkara, ceylonlinux.com

dn: cn=Vishaka Nanayakkara,dc=ceylonlinux,dc=com

objectClass: user

cn: Vishaka Nanayakkara

sn: Nanayakkara

givenName: Vishaka

book1, ceylonlinux.com

dn: documentTitle=book1,dc=ceylonlinux,dc=com

objectClass: document

description: abstract1

documentTitle: book1

documentAuthor: cn=Gihan Dias,dc=ceylonlinux,dc=com

documentAuthor: cn=Shantha Fernando,dc=ceylonlinux,dc=com

documentIdentifier: document 1

book2, ceylonlinux.com

dn: documentTitle=book2,dc=ceylonlinux,dc=com

objectClass: document

description: abstract2

documentTitle: book2

documentAuthor: cn=Shantha Fernando,dc=ceylonlinux,dc=com

documentIdentifier: document 2

search reference

ref: ldap://localhost:9012/dc=ceylonlinux,dc=com??sub

```
# ceylonlinux.com
dn: dc=ceylonlinux,dc=com
objectClass: organization
objectClass: dcObject
o: ceylonlinux
dc: ceylonlinux
```

```
# search result
search: 2
result: 0 Success
```

```
# numResponses: 8
# numEntries: 6
# numReferences: 1
[root@suranga src]#
```

Postfix Configuration

This part describes the method to install postfix from the source code and the method to have virtual mail accounts in maildir format. You have to note that the postfix server used here is not used as a mail gateway. It will be used as a destination mail server where all the user accounts reside. Following section will describe the steps to install postfix from the source code.

Suppose you have copied **postfix-2.2.10.tar.gz** in the **/usr/local/src** directory. As the initial step you have to unzip and then extract the files from the tar bundle.

```
[root@three src]# gzip -d postfix-2.2.10.tar.gz
```

It will extract the files and then you have to use the following command to extract the files from the tar bundle.

```
[root@three src]# tar -xf postfix-2.2.10.tar
```

Then change into the **postfix-2.2.10** directory.

```
[root@three src]# cd postfix-2.2.10
```

Then execute the following command.

```
[root@three postfix-2.2.10]# make
```

Before installing postfix you have to find where the sendmail resides and rename it because otherwise postfix will replace it.

```
[root@three postfix-2.2.10]# whereis sendmail
```

Suppose the location of send mail is **/usr/sbin/**. Then rename it.

```
[root@three postfix-2.2.10]# mv /usr/sbin/sendmail /usr/sbin/sendmail.orig
```

In the same manner rename the existing mailq and newaliases.

```
[root@three postfix-2.2.10]# whereis mailq
```

```
[root@three postfix-2.2.10]# mv /usr/bin/mailq /usr/bin/mailq.orig
```

```
[root@three postfix-2.2.10]# whereis newaliases
```

```
[root@three postfix-2.2.10]# mv /usr/bin/newaliases /usr/bin/newaliases.orig
```

Then you have to create an user account named **postfix** and a group account named **postdrop** that will own postfix.

```
[root@three postfix-2.2.10]#useradd postfix -u 2001
```

```
[root@three postfix-2.2.10]#groupadd postdrop -g 2002
```

After those steps execute the following command while being in the **postfix-2.2.10** directory.

```
[root@three postfix-2.2.10]# make install
```

This step will complete the installation. The following section will guide you on how to create virtual accounts in maildir format.

The virtual domains is **ceylonlinux.com**. Maildir uses directories to store the mails.

First create a base directory to carry the domain directories.

```
[root@three postfix-2.2.10]#mkdir /usr/local/vmail
```

Then create the base directories inside the **vmail** directory

```
[root@three postfix-2.2.10]#cd /usr/local/vmail
```

```
[root@three vmail]#mkdir ceylonlinux.com
```

Then make directories for the users inside each domain directory.

```
[root@three ceylonlinux.com]#mkdir suranga
```

```
[root@three ceylonlinux.com]#mkdir shantha
```

In maildirs you need to have three specific subdirectories inside the user's directory.

```
[root@three ceylonlinux.com]#cd suranga
```

```
[root@three suranga]#mkdir new
```

```
[root@three suranga]#mkdir cur
```

```
[root@three suranga]#mkdir tmp
```

```
[root@three ceylonlinux.com]#cd shantha
```

```
[root@three suranga]#mkdir new
```

```
[root@three suranga]#mkdir cur
```

```
[root@three suranga]#mkdir tmp
```

These mail directories have to be owned by a user and a group. Create that user account and the group account.

```
[root@three ~]#useradd vmail -u 1003
```

```
[root@three ~]#groupadd vmail -u 1005
```

Then change the ownership of the mail directories to those user and group accounts.

```
[root@three ~]#chown vmail /usr/local/vmail
```

These mail directories should have read and write permissions. Hence give those permissions.

```
[root@three ~]#chmod 700 /usr/local/vmail/ceylonlinux.com/suranga/*
```

```
[root@three ~]#chmod 700 /usr/local/vmail/ceylonlinux.com/shantha/*
```

Then edit the configuration file.

```
[root@three postfix-2.2.10]#postconf -e " virtual_mailbox_domains = ceylonlinux.com"
```

This will add an entry about the virtual domains to the configuration file.

```
[root@three postfix-2.2.10]#postconf -e " virtual_mailbox_base = /"
```

The above command will specify the base mail directory. NOTE: Here we do not need to give the virtual_mailbox_base = /usr/local/vmail because this argument is directly coming from ldap server, using ldap attribute in our user schema known as mailMessageStore. [refer my user.schema](ex: /usr/local/vmail/ceylonlinux.com/shantha/)

Then specify the vmail user and the group.

```
[root@three postfix-2.2.10]#postconf -e " virtual_uid_maps = static : 1003"
```

```
[root@three postfix-2.2.10]#postconf -e " virtual_gid_maps = static : 1005"
```

After performing all those steps you can start postfix.

```
[root@three postfix-2.2.10]#postfix start
```

If you make any changes to the configuration file you have to reload postfix.

```
[root@three postfix-2.2.10]#postfix reload
```

To stop postfix

```
[root@three postfix-2.2.10]#postfix stop
```

Anyway, I am adding the main.cf file in the appendix. So you do not need to worry too much. There I did a smart filtering with my own schema that I have build up to communicate with back-sql. You can send me a thank you mail if you can figure out that (suranga at ceylonlinux dot com)

Appendix

Here is the main.cf file in postfix

```
queue_directory = /var/spool/postfix  
command_directory = /usr/sbin  
daemon_directory = /usr/libexec/postfix  
mail_owner = postfix  
myhostname = mail.ceylonlinux.com  
mydomain = ceylonlinux.com  
myorigin = $myhostname  
myorigin = $mydomain  
unknown_local_recipient_reject_code = 550
```

debug_peer_level = 2

debugger_command =

PATH=/bin:/usr/bin:/usr/local/bin:/usr/X11R6/bin

xxgdb \$daemon_directory/\$process_name \$process_id & sleep 5

sendmail_path = /usr/sbin/sendmail

newaliases_path = /usr/bin/newaliases

mailq_path = /usr/bin/mailq

setgid_group = postdrop

html_directory = no

manpage_directory = /usr/local/man

sample_directory = /etc/postfix

readme_directory = no

virtual_mailbox_domains = ceylonlinux.com

####suranga added this part to communicate with LDAP

local_transport = virtual

virtual_mailbox_base = /

virtual_mailbox_maps = ldap:ldapvirtual

virtual_uid_maps = static:1003

virtual_gid_maps = static:1005

ldapvirtual_server_host = localhost

ldapvirtual_scope = sub

ldapvirtual_search_base = dc=ceylonlinux,dc=com

ldapvirtual_query_filter = (&(objectClass=user)(userUserID=%s))

ldapvirtual_result_attribute = mailMessageStore

ldapvirtual_lookup_wildcards = no

alias_maps = ldap:ldapalias

ldapalias_server_host = localhost

ldapalias_search_base = dc=ceylonlinux,dc=com

ldapalias_scope = sub

ldapalias_query_filter = (&(objectClass=user)(userUserID=%s))

ldapalias_result_attribute = mail

Here is the user.schema


```
#
# user.schema by M.B.G.Suranga De Silva
#
# Last Modified Date: Thu Jul 20 2006
# Include it in OpenLDAP's configuration file (slapd.conf)
# and restart slapd.
#

#
# user Attributes Types
#

# userEmail
attributetype ( 1.3.6.1.4.1.19848.1.21.1.1.1 NAME 'userEmail'
    DESC 'Email Address of the user'
    EQUALITY caseIgnoreIA5Match
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )

# userPass
attributetype ( 1.3.6.1.4.1.19848.1.21.1.1.2 NAME 'userPass'
    DESC 'Password of the user'
    EQUALITY caseExactIA5Match
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )

# userFaculty
attributetype ( 1.3.6.1.4.1.19848.1.21.1.1.3 NAME 'userFaculty'
    DESC 'Faculty of the user'
    EQUALITY caseIgnoreMatch
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{10} SINGLE-VALUE )

# userField
attributetype ( 1.3.6.1.4.1.19848.1.21.1.1.4 NAME 'userField'
    DESC 'Field of study of the user'
    EQUALITY caseIgnoreIA5Match
    SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE )

# userFullName
attributetype ( 1.3.6.1.4.1.19848.1.21.1.1.5 NAME 'userFullName'
    DESC 'Name of the user'
```

EQUALITY caseIgnoreIA5Match
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE)

userFirstName

attributetype (1.3.6.1.4.1.19848.1.21.1.1.6 NAME 'userFirstName'
DESC 'First Name of the user'
EQUALITY caseIgnoreMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{16} SINGLE-VALUE)

userLastName

attributetype (1.3.6.1.4.1.19848.1.21.1.1.7 NAME 'userLastName'
DESC 'Last Name of the user'
EQUALITY caseIgnoreIA5Match
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE)

userDateOfBirth

attributetype (1.3.6.1.4.1.19848.1.21.1.1.8 NAME 'userDateOfBirth'
DESC 'user Birthday'
EQUALITY caseIgnoreMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{1} SINGLE-VALUE)

userUserID

attributetype (1.3.6.1.4.1.19848.1.21.1.1.9 NAME 'userUserID'
DESC 'User ID of the user'
EQUALITY caseIgnoreMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{1} SINGLE-VALUE)

userNIC

attributetype (1.3.6.1.4.1.19848.1.21.1.1.10 NAME 'userNIC'
DESC 'NIC of the user'
EQUALITY integerMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE)

userPermanentAddress

attributetype (1.3.6.1.4.1.19848.1.21.1.1.11 NAME 'userPermanentAddress'
DESC 'Permanent Address of the user'
EQUALITY caseExactIA5Match
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE)

userTemporaryAddress

attributetype (1.3.6.1.4.1.19848.1.21.1.1.13 NAME 'userTemporaryAddress'
DESC 'Temporary Address of the user'
EQUALITY caseIgnoreMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{4} SINGLE-VALUE)

userPermanentTelephone

attributetype (1.3.6.1.4.1.19848.1.21.1.1.14 NAME 'userPermanentTelephone'
DESC 'Permanent Telephone Number of the user'
EQUALITY caseIgnoreMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{3} SINGLE-VALUE)

userMobileNumber

attributetype (1.3.6.1.4.1.19848.1.21.1.1.15 NAME 'userMobileNumber'
DESC 'Mobile Number of the user'
EQUALITY caseIgnoreMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{3} SINGLE-VALUE)

userOfficeNumber

attributetype (1.3.6.1.4.1.19848.1.21.1.1.16 NAME 'userOfficeNumber'
DESC 'Office Number of the user'
EQUALITY caseExactMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{3} SINGLE-VALUE)

userAddedBy

attributetype (1.3.6.1.4.1.19848.1.21.1.1.17 NAME 'userAddedBy'
DESC 'The Name of the person who added the user'
EQUALITY caseExactMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{1} SINGLE-VALUE)

userAddedDate

attributetype (1.3.6.1.4.1.19848.1.21.1.1.18 NAME 'userAddedDate'
DESC 'Added Date'
EQUALITY integerMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE)

userLastModifiedBy

attributetype (1.3.6.1.4.1.19848.1.21.1.1.19 NAME 'userLastModifiedBy'
DESC 'Name of the person who modified user Last'

EQUALITY integerMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE)

userLevel

attributetype (1.3.6.1.4.1.19848.1.21.1.1.20 NAME 'userLevel'
DESC 'Level'
EQUALITY integerMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE)

userLastModifiedDate

attributetype (1.3.6.1.4.1.19848.1.21.1.1.21 NAME 'userLastModifiedDate'
DESC 'Last Modified Date'
EQUALITY integerMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE)

userOfficeAddress

attributetype (1.3.6.1.4.1.19848.1.21.1.1.22 NAME 'userOfficeAddress'
DESC 'Office Address'
EQUALITY integerMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE)

userAlternateEmail

attributetype (1.3.6.1.4.1.19848.1.21.1.1.23 NAME 'userAlternateEmail'
DESC 'Alternate Email Address'
EQUALITY integerMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE)

userActiveStatus

attributetype (1.3.6.1.4.1.19848.1.21.1.1.24 NAME 'userActiveStatus'
DESC 'Status of the user'
EQUALITY caseIgnoreIA5Match
SUBSTR caseIgnoreSubstringsMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 SINGLE-VALUE)

userGender

attributetype (1.3.6.1.4.1.19848.1.21.1.1.25 NAME 'userGender'
DESC 'user Gender'
EQUALITY integerMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE)

```
# userRegistrationDate
attributetype ( 1.3.6.1.4.1.19848.1.21.1.1.26 NAME 'userRegistrationDate'
  DESC 'Registration Date of the user'
  EQUALITY booleanMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.7 SINGLE-VALUE )
```

```
# userEmailDir
attributetype ( 1.3.6.1.4.1.19848.1.21.1.1.27 NAME 'userEmailDir'
  DESC 'Email Directory of the user'
  EQUALITY integerMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE )
```

```
#userSpecilization
attributetype ( 1.3.6.1.4.1.19848.1.21.1.1.28 NAME 'userSpecilization'
  DESC 'Specilization of the user'
  EQUALITY integerMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE )
```

```
#mailMessageStore
attributetype ( 1.3.6.1.4.1.19848.1.21.1.1.29 NAME 'mailMessageStore'
  DESC 'mailMessageStore of the user'
  EQUALITY caseIgnoreMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{10} SINGLE-VALUE )
```

```
#attributetype ( 1.3.6.1.4.1.19848.1.21.1.1.30 NAME 'userid'
#   DESC 'User id of the user'
#   EQUALITY caseIgnoreMatch
#   SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{10} SINGLE-VALUE )
```

```
#
# user Object Class
#
```

```
# user
objectclass ( 1.3.6.1.4.1.19848.1.21.1.1
  NAME 'user'
```

```
SUP top
DESC 'Information of the user'
AUXILIARY
MAY ( userUserID $ userPass $ userEmail $ userFaculty $ userFullName $ userField $
userFirstName $
      userLastName $ userDateOfBirth $ userNIC $ userPermanentAddress $
      userTemporaryAddress $ userPermanentTelephone $ userMobileNumber $
userOfficeNumber $
      userAddedBy $ userAddedDate $ userLastModifiedBy $ userLevel $
      userLastModifiedDate $ userOfficeAddress $ userAlternateEmail $ userActiveStatus $
      userGender $ userRegistrationDate $ userEmailDir $ userSpecilization $
mailMessageStore ))
```

Here is the slapd.conf

```
#
# See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.
#
include      /usr/local/etc/openldap/schema/core.schema
include      /usr/local/etc/openldap/schema/cosine.schema
include      /usr/local/etc/openldap/schema/inetorgperson.schema
include      /usr/local/etc/openldap/schema/nis.schema
include      /usr/local/etc/openldap/schema/corba.schema
include      /usr/local/etc/openldap/schema/java.schema
#include     /usr/local/etc/openldap/schema/kerberosobject.schema
#include     /usr/local/etc/openldap/schema/openldap.schema
#include     /usr/local/etc/openldap/schema/qmail.schema
include      /usr/local/etc/openldap/schema/user.schema

# Define global ACLs to disable default read access.
access to *
    by * write

# Do not enable referrals until AFTER you have a working directory
# service AND an understanding of referrals.
#referral    ldap://root.openldap.org

allow bind_v2

pidfile      /usr/local/var/run/slapd.pid
argsfile     /usr/local/var/run/slapd.args
```

schemacheck on

#####

sql database definitions

#####

```
database    sql
suffix      "dc=ceylonlinux,dc=com"
rootdn      "cn=root,dc=ceylonlinux,dc=com"
#rootpw     {SSHA}X59lpHb78L7JaNunf7J12FyAMJgeaOdD
rootpw      secret
dbname      ldap
dbuser      ldap
dbpasswd    password
#insentry_query "insert into ldap_entries (id,dn,oc_map_id,parent,keyval) values ((select max(id)+1
from ldap_entries),?,?,?,?)"
subtree_cond "ldap_entries.dn LIKE CONCAT('%',?)"
insentry_query "INSERT INTO ldap_entries (dn,oc_map_id,parent,keyval) VALUES (?,?,?,?)"
#upper_func  "upper"
#strcast_func "text"
#concat_pattern "?||?"
has_ldapinfo_dn_ru no

#lastmod off
```


Here is the backsql create.sql

```
drop table if exists ldap_oc_mappings;
create table ldap_oc_mappings
(
    id integer unsigned not null primary key auto_increment,
    name varchar(64) not null,
    keytbl varchar(64) not null,
    keycol varchar(64) not null,
    create_proc varchar(255),
    delete_proc varchar(255),
    expect_return tinyint not null
);

drop table if exists ldap_attr_mappings;
create table ldap_attr_mappings
(
    id integer unsigned not null primary key auto_increment,
    oc_map_id integer unsigned not null references ldap_oc_mappings(id),
    name varchar(255) not null,
    sel_expr varchar(255) not null,
    sel_expr_u varchar(255),
    from_tbls varchar(255) not null,
    join_where varchar(255),
    add_proc varchar(255),
    delete_proc varchar(255),
```

```

        param_order tinyint not null,
        expect_return tinyint not null
    );

drop table if exists ldap_entries;
create table ldap_entries
(
    id integer unsigned not null primary key auto_increment,
    dn varchar(255) not null,
    oc_map_id integer unsigned not null references ldap_oc_mappings(id),
    parent int NOT NULL ,
    keyval int NOT NULL
);

alter table ldap_entries add
    constraint unq1_ldap_entries unique
(
    oc_map_id,
    keyval
);

alter table ldap_entries add
    constraint unq2_ldap_entries unique
(
    dn
);

drop table if exists ldap_entry_objclasses;
create table ldap_entry_objclasses
(
    entry_id integer not null references ldap_entries(id),
    oc_name varchar(64)
);

```

Here is backsql drop.sql

```
DROP TABLE IF EXISTS ldap_entry_objclasses;
```

```
DROP TABLE IF EXISTS ldap_attr_mappings;
```

```
DROP TABLE IF EXISTS ldap_entries;
```

```
DROP TABLE IF EXISTS ldap_oc_mappings;
```

Here is testdb create ceylonlinux.sql

```
drop table if exists persons;
```

```
CREATE TABLE persons (  
    id int NOT NULL,  
    name varchar(255) NOT NULL,  
    surname varchar(255) NOT NULL,  
    password varchar(64),  
    faculty varchar(64),  
    maildir varchar(250),  
    mail varchar(64),  
    userid varchar(64)
```

```
);
```

```
drop table if exists institutes;
```

```
CREATE TABLE institutes (  
    id int NOT NULL,  
    name varchar(255)
```

```
);
```

```
drop table if exists documents;
```

```
CREATE TABLE documents (  
    id int NOT NULL,  
    title varchar(255) NOT NULL,  
    abstract varchar(255)
```

```
);
```

```
drop table if exists authors_docs;
```

```
CREATE TABLE authors_docs (  
    pers_id int NOT NULL,  
    doc_id int NOT NULL
```

```
);
```

```
drop table if exists phones;
```

```
CREATE TABLE phones (  
    id int NOT NULL ,  
    phone varchar(255) NOT NULL ,  
    pers_id int NOT NULL
```

```
);
```

```
ALTER TABLE authors_docs ADD  
    CONSTRAINT PK_authors_docs PRIMARY KEY  
    (  
        pers_id,  
        doc_id  
    );
```

```
ALTER TABLE documents ADD  
    CONSTRAINT PK_documents PRIMARY KEY
```

```
(
    id
);
```

```
ALTER TABLE institutes ADD
    CONSTRAINT PK_institutes PRIMARY KEY
(
    id
);
```

```
ALTER TABLE persons ADD
    CONSTRAINT PK_persons PRIMARY KEY
(
    id
);
```

```
ALTER TABLE phones ADD
    CONSTRAINT PK_phones PRIMARY KEY
(
    id
);
```

```
drop table if exists referrals;
CREATE TABLE referrals (
    id int NOT NULL,
    name varchar(255) NOT NULL,
    url varchar(255) NOT NULL
);
```

---This part is added by suranga

```
CREATE TABLE ceylonlinuxuser (
    id int NOT NULL,
    userPassword varchar(20),
    userFaculty varchar(32) default NULL,
    userField varchar(20),
    userFullName varchar(20),
```

```
userFirstName varchar(20),
userLastName varchar(20),
userDateOfBirth varchar(20),
userUserID varchar(20),
userNIC varchar(20),
userPermanentAddress varchar(20),
userTemporaryAddress varchar(20),
userPermanentTelephone varchar(20),
userMobileNumber varchar(20),
userOfficeNumber varchar(20),
userAddedBy varchar(20),
userAddedDate varchar(20),
userLastModifiedBy varchar(20),
userLevel varchar(20),
userLastModifiedDate varchar(20),
userOfficeAddress varchar(20),
userAlternateEmail varchar(20),
userActiveStatus varchar(20),
userGender varchar(20),
userRegistrationDate varchar(20),
userEmailDir varchar(20),
userSpecilization varchar(20)
);
```

```
ALTER TABLE ceylonlinuxuser ADD
CONSTRAINT PK_oumuser PRIMARY KEY
(
    id
);
```

-----end addition

Here is testdb data ceylonlinux.sql

```
insert into institutes (id,name) values (1,'ceylonlinux');
```

```
insert into persons (id,name,surname,password,faculty,maildir,mail,userid) values  
(1,'Shantha','Feranado','mit','cse','/usr/local/vmail/ceylonlinux.com/shantha','shantha@ceylonlinux.com',  
'shantha');
```

```
insert into persons (id,name,surname) values (2,'Gihan','Dias');
```

```
insert into persons (id,name,surname) values (3,'Vishaka','Nanayakkara');
```

```
insert into persons (id,name,surname) values (4,'Gayan','Suranga');
```

```
insert into phones (id,phone,pers_id) values (1,'332-2334',1);
```

```
insert into phones (id,phone,pers_id) values (2,'222-3234',1);
```

```
insert into phones (id,phone,pers_id) values (3,'545-4563',2);
```

```
insert into documents (id,abstract,title) values (1,'abstract1','book1');
```

```
insert into documents (id,abstract,title) values (2,'abstract2','book2');
```

```
insert into authors_docs (pers_id,doc_id) values (1,1);
```

```
insert into authors_docs (pers_id,doc_id) values (1,2);
```

```
insert into authors_docs (pers_id,doc_id) values (2,1);
```

```
insert into referrals (id,name,url) values (1,'Referral','ldap://localhost:9012/');
```

```
---this part is added by suranga
```

```
insert into ceylonlinuxuser (id,userFaculty,userField,userFullName) values (1,'IT','ENG','Suranga De  
Silva');
```

```
-----end addition
```

Here is testdb drop ceylonlinux.sql

```
DROP TABLE IF EXISTS persons;
```

```
DROP TABLE IF EXISTS institutes;
```

```
DROP TABLE IF EXISTS documents;
DROP TABLE IF EXISTS authors_docs;
DROP TABLE IF EXISTS phones;
DROP TABLE IF EXISTS ceylonlinuxuser;
```

Here is the most important testdb metadata ceylonlinux.sql

```
-- mappings
```

```
-- objectClass mappings: these may be viewed as structuralObjectClass, the ones that are used to decide how to build an entry
```

```
-- id          a unique number identifying the objectClass
-- name        the name of the objectClass; it MUST match the name of an objectClass that is loaded in slapd's schema
-- keytbl      the name of the table that is referenced for the primary key of an entry
-- keycol      the name of the column in "keytbl" that contains the primary key of an entry; the pair "keytbl.keycol" uniquely identifies an entry of objectClass "id"
-- create_proc a procedure to create the entry
-- delete_proc a procedure to delete the entry; it takes "keytbl.keycol" of the row to be deleted
-- expect_return a bitmap that marks whether create_proc (1) and delete_proc (2) return a value or not
```

```
insert into ldap_oc_mappings (id,name,keytbl,keycol,create_proc,delete_proc,expect_return)
values (1,'user','persons','id',NULL,NULL,0);
```

```
insert into ldap_oc_mappings (id,name,keytbl,keycol,create_proc,delete_proc,expect_return)
values (2,'document','documents','id',NULL,NULL,0);
```

```
insert into ldap_oc_mappings (id,name,keytbl,keycol,create_proc,delete_proc,expect_return)
values (3,'organization','institutes','id',NULL,NULL,0);
```

```
insert into ldap_oc_mappings (id,name,keytbl,keycol,create_proc,delete_proc,expect_return)
```



```
values (4,'referral','referrals','id',NULL,NULL,0);
```

```
----suranga adding
```

```
--insert into ldap_oc_mappings (id,name,keytbl,keycol,create_proc,delete_proc,expect_return)
--values (5,'student','persons','id',NULL,NULL,0);
```

```
----end addition
```

```
-- attributeType mappings: describe how an attributeType for a certain objectClass maps to the SQL
data.
```

```
-- id a unique number identifying the attribute
```

```
-- oc_map_id the value of "ldap_oc_mappings.id" that identifies the objectClass this
attributeType is defined for
```

```
-- name the name of the attributeType; it MUST match the name of an attributeType that
is loaded in slapd's schema
```

```
-- sel_expr the expression that is used to select this attribute (the "select <sel_expr> from ..."
portion)
```

```
-- from_tbls the expression that defines the table(s) this attribute is taken from (the "select ...
from <from_tbls> where ..." portion)
```

```
-- join_where the expression that defines the condition to select this attribute (the "select ...
where <join_where> ..." portion)
```

```
-- add_proc a procedure to insert the attribute; it takes the value of the attribute that is added,
and the "keytbl.keycol" of the entry it is associated to
```

```
-- delete_proc a procedure to delete the attribute; it takes the value of the attribute that is added,
and the "keytbl.keycol" of the entry it is associated to
```

```
-- param_order a mask that marks if the "keytbl.keycol" value comes before or after the value in
add_proc (1) and delete_proc (2)
```

```
-- expect_return a mask that marks whether add_proc (1) and delete_proc(2) are expected to
return a value or not
```

```
insert into ldap_attr_mappings
```

```
(id,oc_map_id,name,sel_expr,from_tbls,join_where,add_proc,delete_proc,param_order,expect_return)
```

```
values (1,1,'cn',"concat(persons.name,' ',persons.surname)",'persons',NULL,NULL,NULL,3,0);
```

```
insert into ldap_attr_mappings
```

```
(id,oc_map_id,name,sel_expr,from_tbls,join_where,add_proc,delete_proc,param_order,expect_return)
```

```
values (2,1,'telephoneNumber','phones.phone','persons,phones',
```

```
'phones.pers_id=persons.id',NULL,NULL,3,0);
```

```
insert into ldap_attr_mappings
```

```
(id,oc_map_id,name,sel_expr,from_tbls,join_where,add_proc,delete_proc,param_order,expect_return)
values (3,1,'givenName','persons.name','persons',NULL,NULL,NULL,3,0);
```

```
insert into ldap_attr_mappings
```

```
(id,oc_map_id,name,sel_expr,from_tbls,join_where,add_proc,delete_proc,param_order,expect_return)
values (4,1,'sn','persons.surname','persons',NULL,NULL,NULL,3,0);
```

```
insert into ldap_attr_mappings
```

```
(id,oc_map_id,name,sel_expr,from_tbls,join_where,add_proc,delete_proc,param_order,expect_return)
values (5,1,'userPassword','persons.password','persons','persons.password IS NOT
NULL',NULL,NULL,3,0);
```

```
-----suranga is adding
```

```
insert into ldap_attr_mappings
```

```
(id,oc_map_id,name,sel_expr,from_tbls,join_where,add_proc,delete_proc,param_order,expect_return)
values (15,1,'userFaculty','persons.faculty','persons',NULL,NULL,NULL,3,0);
```

```
insert into ldap_attr_mappings
```

```
(id,oc_map_id,name,sel_expr,from_tbls,join_where,add_proc,delete_proc,param_order,expect_return)
values (16,1,'mailMessageStore','persons.maildir','persons',NULL,NULL,NULL,3,0);
```

```
insert into ldap_attr_mappings
```

```
(id,oc_map_id,name,sel_expr,from_tbls,join_where,add_proc,delete_proc,param_order,expect_return)
values (17,1,'mail','persons.mail','persons',NULL,NULL,NULL,3,0);
```

```
insert into ldap_attr_mappings
```

```
(id,oc_map_id,name,sel_expr,from_tbls,join_where,add_proc,delete_proc,param_order,expect_return)
values (18,1,'userUserID','persons.userid','persons',NULL,NULL,NULL,3,0);
```

```
----end adding
```

```
insert into ldap_attr_mappings
```

```
(id,oc_map_id,name,sel_expr,from_tbls,join_where,add_proc,delete_proc,param_order,expect_return)
values (6,1,'seeAlso','seeAlso.dn','ldap_entries AS seeAlso,documents,authors_docs,persons',
```

```
'seeAlso.keyval=documents.id AND seeAlso.oc_map_id=2 AND
authors_docs.doc_id=documents.id AND authors_docs.pers_id=persons.id',
    NULL,NULL,3,0);
```

```
insert into ldap_attr_mappings
(id,oc_map_id,name,sel_expr,from_tbls,join_where,add_proc,delete_proc,param_order,expect_return)
values (7,2,'description','documents.abstract','documents',NULL,NULL,NULL,3,0);
```

```
insert into ldap_attr_mappings
(id,oc_map_id,name,sel_expr,from_tbls,join_where,add_proc,delete_proc,param_order,expect_return)
values (8,2,'documentTitle','documents.title','documents',NULL,NULL,NULL,3,0);
```

```
insert into ldap_attr_mappings
(id,oc_map_id,name,sel_expr,from_tbls,join_where,add_proc,delete_proc,param_order,expect_return)
values (9,2,'documentAuthor','documentAuthor.dn','ldap_entries AS
documentAuthor,documents,authors_docs,persons',
    'documentAuthor.keyval=persons.id AND documentAuthor.oc_map_id=1 AND
authors_docs.doc_id=documents.id AND authors_docs.pers_id=persons.id',
    NULL,NULL,3,0);
```

```
insert into ldap_attr_mappings
(id,oc_map_id,name,sel_expr,from_tbls,join_where,add_proc,delete_proc,param_order,expect_return)
values (10,2,'documentIdentifier','concat("document
",documents.id)','documents',NULL,NULL,NULL,3,0);
```

```
insert into ldap_attr_mappings
(id,oc_map_id,name,sel_expr,from_tbls,join_where,add_proc,delete_proc,param_order,expect_return)
values (11,3,'o','institutes.name','institutes',NULL,NULL,NULL,3,0);
```

```
insert into ldap_attr_mappings
(id,oc_map_id,name,sel_expr,from_tbls,join_where,add_proc,delete_proc,param_order,expect_return)
values (12,3,'dc','lower(institutes.name)','institutes,ldap_entries AS dcObject,ldap_entry_objclasses as
auxObjectClass',
    'institutes.id=dcObject.keyval AND dcObject.oc_map_id=3 AND
dcObject.id=auxObjectClass.entry_id AND auxObjectClass.oc_name="dcObject"',
    NULL,NULL,3,0);
```

```
insert into ldap_attr_mappings
(id,oc_map_id,name,sel_expr,from_tbls,join_where,add_proc,delete_proc,param_order,expect_return)
```

```
values (13,4,'ou','referrals.name','referrals',NULL,NULL,NULL,3,0);
```

```
insert into ldap_attr_mappings
```

```
(id,oc_map_id,name,sel_expr,from_tbls,join_where,add_proc,delete_proc,param_order,expect_return)
```

```
values (14,4,'ref','referrals.url','referrals',NULL,NULL,NULL,3,0);
```

```
-- entries mapping: each entry must appear in this table, with a unique DN rooted at the database  
naming context
```

```
-- id          a unique number > 0 identifying the entry
```

```
-- dn          the DN of the entry, in "pretty" form
```

```
-- oc_map_id   the "ldap_oc_mappings.id" of the main objectClass of this entry (view it as the  
structuralObjectClass)
```

```
-- parent     the "ldap_entries.id" of the parent of this objectClass; 0 if it is the "suffix" of the  
database
```

```
-- keyval     the value of the "keytbl.keycol" defined for this objectClass
```

```
insert into ldap_entries (id,dn,oc_map_id,parent,keyval)
```

```
values (1,'dc=ceylonlinux,dc=com',3,0,1);
```

```
insert into ldap_entries (id,dn,oc_map_id,parent,keyval)
```

```
values (2,'cn=Shantha Fernando,dc=ceylonlinux,dc=com',1,1,1);
```

```
insert into ldap_entries (id,dn,oc_map_id,parent,keyval)
```

```
values (3,'cn=Gihan Dias,dc=ceylonlinux,dc=com',1,1,2);
```

```
insert into ldap_entries (id,dn,oc_map_id,parent,keyval)
```

```
values (4,'cn=Vishaka Nanayakkara,dc=ceylonlinux,dc=com',1,1,3);
```

```
insert into ldap_entries (id,dn,oc_map_id,parent,keyval)
```

```
values (5,'documentTitle=book1,dc=ceylonlinux,dc=com',2,1,1);
```

```
insert into ldap_entries (id,dn,oc_map_id,parent,keyval)
```

```
values (6,'documentTitle=book2,dc=ceylonlinux,dc=com',2,1,2);
```

```
insert into ldap_entries (id,dn,oc_map_id,parent,keyval)
```

```
values (7,'ou=Referral,dc=ceylonlinux,dc=com',4,1,1);
```

```
-- objectClass mapping: entries that have multiple objectClass instances are listed here with the  
objectClass name (view them as auxiliary objectClass)
```

```
-- entry_id   the "ldap_entries.id" of the entry this objectClass value must be added
```

-- oc_name the name of the objectClass; it MUST match the name of an objectClass
that is loaded in slapd's schema

```
insert into ldap_entry_objclasses (entry_id,oc_name)  
values (1,'dcObject');
```

```
insert into ldap_entry_objclasses (entry_id,oc_name)  
values (7,'extensibleObject');
```

IMAP and POP3 using dovecot with openldap

- configure dovecot –with-ldap support and install
- change dovecot.conf and dovecot-ldap.conf according to the above scenario
- Here are the configuration files

Here is my dovecot.conf

```
protocols = imap imaps pop3 pop3s
first_valid_uid = 1003
last_valid_uid = 1003
auth_mechanisms = plain digest-md5
auth_executable = /usr/libexec/dovecot/dovecot-auth
auth_user = dovecot-auth
disable_plaintext_auth = no
ssl_disable = yes
login_dir = /var/run/dovecot/login
login_user = dovecot
default_mail_env = maildir:/usr/local/vmail/%d/%n
mail_debug = yes
mail_log_prefix = "%Us(%u): "
verbose_proctitle = yes

protocol imap {
}
```

```
protocol pop3 {  
login_executable = /usr/libexec/dovecot/pop3-login  
mail_executable = /usr/libexec/dovecot/pop3  
pop3_uidl_format = %08Xu%08Xv  
mail_plugin_dir = /usr/lib/dovecot/pop3  
}
```

```
protocol lda {  
  postmaster_address = postmaster@ceylonlinux.com  
}
```

```
auth_executable = /usr/libexec/dovecot/dovecot-auth  
auth_process_size = 256  
auth_username_chars =  
abcdefghijklmnopqrstuvxyzABCDEFGHIJKLMNOPQRSTUVWXYZ01234567890.-_@  
auth_verbose = yes  
auth_debug_passwords = yes
```

```
auth default {  
  mechanisms = plain login
```

```
  passdb pam {  
  
  }
```

```
  passdb ldap {  
    args = /etc/dovecot-ldap.conf  
  }
```

```
  userdb ldap {  
    args = /etc/dovecot-ldap.conf  
  }
```

```
  user = root  
  
}
```

```
dict {  
  
}  
  
plugin {  
  
}
```

Here is my dovecot-ldap.conf file

```
hosts = localhost
```

```
dn = cn=root,dc=ceylonlinux,dc=com
```

```
# Password for LDAP server
```

```
dnpass = secret
```

```
ldap_version = 3
```

```
base = dc=ceylonlinux,dc=com
```

```
deref = never
```

```
scope = subtree
```

```
# User attributes are given in LDAP-name=dovecot-internal-name list. The
```

```
# internal names are:
```

```
# home: Home directory
```

```
# mail: MAIL environment
```

```
# system_user: System user name (for getting user's groups from /etc/group)
```

```
# - For virtual users you don't want to use this, so this defaults to none.
```



```

# uid: System user ID
# gid: System group ID
#user_attrs = mail,homeDirectory=home,uidNumber=uid,gidNumber=gid
user_attrs = mail,homeDirectory=mailMessageStore,uidNumber=1003,gidNumber=1003

# Filter for user lookup. Some variables can be used (see
# http://wiki.dovecot.org/Variables for full list):
# %u - username
# %n - user part in user@domain, same as %u if there's no domain
# %d - domain part in user@domain, empty if user there's no domain
user_filter = (&(objectClass=user)(mail=%u))

# Password checking attributes:
# user: Virtual user name (user@domain), if you wish to change the
# user-given username to something else
# password: Password, may optionally start with {type}, eg. {crypt}
pass_attrs = mail=user,userPassword=password

# If you wish to avoid two LDAP lookups (passdb + userdb), you can use
# userdb prefetch instead of userdb ldap in dovecot.conf. In that case you'll
# also have to include user_attrs in pass_attrs field prefixed with "userdb_"
# string. For example:
#pass_attrs =
uid=user,userPassword=password,homeDirectory=userdb_home,uidNumber=userdb_uid,gidNumber=
userdb_gid

# Filter for password lookups
pass_filter = (&(objectClass=user)(mail=%u))

# Default password scheme. "{scheme}" before password overrides this.
# List of supported schemes is in: http://wiki.dovecot.org/Authentication
default_pass_scheme = CRYPT

# You can use same UID and GID for all user accounts if you really want to.
# If the UID/GID is still found from LDAP reply, it overrides these values.
user_global_uid = 1003
user_global_gid = 1003

```