



# Address Book X LDAP

Version 1.1  
Revision 4



<b>Foreword</b>	<b>4</b>
<b>Installation Instructions</b>	<b>5</b>
<b>Backup existing Address Book</b>	<b>5</b>
<b>Pre-requisites</b>	<b>5</b>
<b>OpenLDAP server configuration</b>	<b>5</b>
<i>Generating hashed password</i>	<i>5</i>
<i>Server configuration</i>	<i>5</i>
<i>Schema extension (Optional - but recommended)</i>	<i>6</i>
<i>Starting the service</i>	<i>6</i>
<i>Populating the directory</i>	<i>8</i>
<i>Recommended Reading</i>	<i>10</i>
<b>Startup Items</b>	<b>11</b>
<i>Instructions for Tiger</i>	<i>11</i>
<i>Option 1 - Launch Daemon</i>	<i>11</i>
<i>Option 2 - Startup Items (not recommended)</i>	<i>11</i>
<i>Instructions for Panther</i>	<i>12</i>
<b>Address Book configuration</b>	<b>13</b>
<i>Mail configuration</i>	<i>13</i>
<b>Address Book X LDAP</b>	<b>14</b>
<i>ABxLDAP Preference Panel</i>	<i>14</i>
<i>Address Book 4 LDAP (v2)</i>	<i>14</i>
<b>Address Book 2 LDAP and Address Book 4 LDAP</b>	<b>15</b>
<b>Extending schema mappings</b>	<b>16</b>
<b>Uninstalling</b>	<b>16</b>
<b>Thunderbird Address Book</b>	<b>17</b>
<i>Connection Configuration</i>	<i>17</i>
<i>Schema Configuration</i>	<i>17</i>

<b>OS X Server Installation</b>	<b>19</b>
<b>Appendix A : SSL Configuration</b>	<b>24</b>
<i>Obtaining a SSL certificate</i>	<i>24</i>
<i>Createing a certificate authority</i>	<i>24</i>
<b>Creating a Certificate</b>	<b>25</b>
<i>Certificate files</i>	<i>27</i>
<i>Configuration of OpenLDAP directory</i>	<i>27</i>
<b>Client Configuration</b>	<b>28</b>
<b>Warning - Address Book &amp; SSL Problem !!!</b>	<b>29</b>
<b>Appendix B : Client Authentication</b>	<b>29</b>
<b>Feedback</b>	<b>30</b>

## Foreword

It all started out with Address Book 2 LDAP which simply transfers all contacts to a LDAP directory. Being my first project on the Mac it lacked a lot of features and polish. With Address Book 4 LDAP the ability to select individual contacts and groups was added. It also provides the capability to create, modify and delete contacts directly from the directory. Between Address Book 4 LDAP and the next iteration, Address Book X LDAP, almost a year past. The the release of Tiger (10.4) the Sync API was now available. The LDAP integration layer was completely rewritten. Previously, in Address Book 2 LDAP and Address Book 4 LDAP, it was implemented in Java and invoked via the Java Bridge. It has now been replaced with native C implementation provided by the LDAP framework. The user interface has been redesigned using Cocoa binding.

Before the release of Address Book X LDAP the software was distributed for free with the option of users making donations to express their support for the project. Most feedback received was positive and encouraged me to continue the development. With the latest release this has been changed and the software is available for £15.00 GBP for a site license and not tied to the number of users. Since this change I found myself spending much more time on development and enhancements and it has made a significant impact on the project, with the biggest being motivational, rather than financial. Having users express their appreciation of ones efforts, with a small contribution, has increased productivity significantly.

Many of the improvements are direct results of suggestions made by the user community. The problem of sharing contact information is a very diverse one, and has multiple solutions. Certain users prefer to store all contacts in a central place, while other would prefer keeping copies on each client. The requirement to share a subset of contact has been put forward on several occasions. I hope to continue in this trend of community involvement in future.

## Installation Instructions

In this section the installation and configuration of and OpenLDAP directory is covered.

## Backup existing Address Book

So far nobody lost their Address Book due to ABxLDAP, but it's better to be safe than sorry. It's a good idea to take a backup of your current Address Book. There is also an automatic backup feature, which can come in useful from time to time for just in case.

<http://www.hawkwings.net/2007/01/20/how-to-recover-missing-address-book-data/>

## Pre-requisites

Root / Administrator account must be enabled. This can be done by going to Go > Utilities > Directory Utility.app, then Edit > Enable Root. Type in your usual password for your account. Then Apple > Log Out and sign back in under Other. For name use "root" and password as that which you just gave in the Directory Utility.app. This will put you into the computer as root user. You can then go to Go > Utilities > Terminal.app. The sudo option is of course available, but for simplicity it might be best to enable the root account.

## OpenLDAP server configuration

The software to run an OpenLDAP directory is already included with Mac OS X and Mac OS X Server. The server edition comes with some nice utilities to configure Open Directory. This section covers the "sticks and stone" approach to getting OpenLDAP running on the non-server edition.

### Generating hashed password

The LDAP administrator password is stored in its hashed form in the configuration file. To generate the hash code for the password the **slappasswd** command is used. Copy and paste the hash code into the configuration file shown later on.

```
wolf:~/Resources/LDAP alex$ slappasswd
New password: [secret ENTER]
Re-enter new password: [secret ENTER]
{SSHA}f1z7UHHB0I+iRVcfm21qaehcokcUj03m
wolf:~/Resources/LDAP alex$
```

### Server configuration

The server configuration file (/etc/openldap/slapd.conf) is shown in the example below, and specifies amongst other parameters which schema are included, the database type and the ldap administrator account. The **rootdn** and **rootpw** parameters define the account which has full access to the directory. The rootpw should contain the password generated in previous section using slappasswd. The following configuration provides is very basic, and does not take security into consideration. I suggest you take a look at the OpenLDAP configuration documentation if you going to put your server on the internet. In one of the later sections you will find instructions on using SSL to encrypt the connection and protect your information from prying eyes.

/etc/openldap/slapd.conf

```
include /etc/openldap/schema/core.schema
include /etc/openldap/schema/cosine.schema
include /etc/openldap/schema/nis.schema
```

```

include          /etc/openldap/schema/inetorgperson.schema
include          /etc/openldap/schema/misc.schema
include          /etc/openldap/schema/samba.schema
include          /etc/openldap/schema/apple.schema
include          /etc/openldap/schema/netinfo.schema
include          /etc/openldap/schema/abldap.schema

pidfile          /var/run/openldap/slapd.pid
argsfile         /var/run/openldap/slapd.args
database         bdb

suffix           "o=j2anywhere,c=gb"
rootdn           "cn=ldapadmin,o=j2anywhere,c=gb"
rootpw           {SSHA}flz7UHHB0I+iRVcfm21qaehcokcUj03m
directory        /var/db/openldap/openldap-data/
index            objectClass      eq

```

The database directory **/var/db/openldap/openldap-data/** must exist and might have to be created. To create this directory use the following command:

```
mkdir /var/db/openldap/openldap-data/
```

### Schema extension (Optional - but recommended)

Address Book X LDAP has support for multiple attribute mappings. To use the preferred mapping options Address Book X LDAP Person rather than InetOrgPerson a new schema extension has to be included. To include the extension copy the abldap.schema schema extension to the **/etc/openldap/schema/** folder and include it as shown in the example above.

### Starting the service

The configuration of the directory is now completed and we are ready to start the directory. For now we are going to run ldap from a root shell, which means you must not close the terminal window once the directory started. The configuration for the directory to start on system start-up is covered in one of the later sections. For now just run the following command as root user :

```
/usr/libexec/slapd -d 255
```

After a lot of text you should see :

```

slapd startup: initiated.
backend_startup: starting "o=j2anywhere,c=gb"
bdb_db_open: o=j2anywhere,c=gb
bdb_db_open: dbenv_open(/var/db/openldap/openldap-data)
slapd starting
daemon: added 7r
daemon: added 8r
daemon: select: listen=7 active_threads=0 tvp=NULL
daemon: select: listen=8 active_threads=0 tvp=NULL

```

At this point LDAP is up and running. Open up a new terminal window and continue the installation process.



## Populating the directory

Now we need to populate the base structure of the directory. To do this we create an basic text file using the LDIF file format. The LDIF format is very specific about it's format, please ensure that there are not extra spaces at the end of the file as this will result in errors during the import process.

### Sample LDIF file (InitialImport.ldif)

```
# j2anywhere, gb
dn: o=j2anywhere,c=gb
objectClass: organization
o: j2anywhere

# people, j2anywhere, gb
dn: ou=people,o=j2anywhere,c=gb
objectClass: organizationalUnit
ou: people
```

If you like you can also include the following sample within the LDIF file to construct a sample contact for testing. Feel free to use your own details.

```
# Alexander Hartner, people, j2anywhere, gb
dn: cn=Alexander Hartner,ou=people,o=j2anywhere,c=gb
objectClass: top
objectClass: person
objectClass: inetOrgPerson
objectClass: abxldapPerson
displayName: Alexander Hartner
cn: Alexander Hartner
givenName: Alexander
sn: Hartner
mail: alex@j2anywhere.com
initials: A
o: j2anywhere.com
```

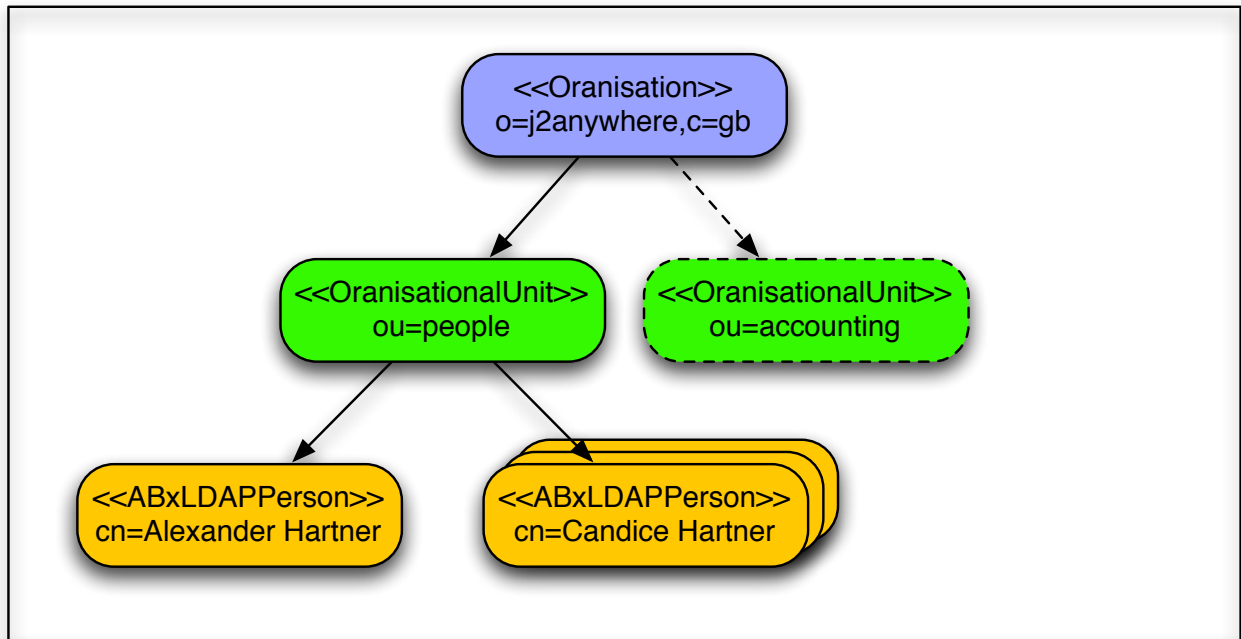
The LDAP directory contains a hierarchy of objects. Each object is of a particular class. The class of the object determines it's attributes. Each class has required or mandatory attributes, which must be provided and optional ones, which may or may not be there. The definition for the attributes and classes resides in the schema files.

The **suffix** from the configuration file specifies the root of the tree. From there onwards the tree is constructed along a given criteria. In the example the tree of constructed around a organisation. The tree then distinguishes by department or organisational unit.

The structure can be extended to contain other departments or categories. The example shows should be sufficient for home and SOHO's usage. For larger organisation refer to the OpenLDAP Administrators guide.

This structure shows the objects created from the ldif file above.





The password used to generate the hashed password which you cut & pasted into the `slapd.conf` file must match the `-w` parameter in the `ldapadd` command.

```
wolf:~/Resources/LDAP alex$ ldapadd -c -D "cn=ldapadmin,o=j2anywhere,c=gb" -w
secret -x -f InitialImport.ldif
...
adding new entry "o=j2anywhere,c=gb"
adding new entry "ou=people,o=j2anywhere,c=gb"
adding new entry "cn=Alexander Hartner,ou=people,o=j2anywhere,c=gb"
...
wolf:~/Resources/LDAP alex$
```

Once the import has been completed you can search ldap with the following command. The `-D` parameter should be the rootdn while the `-b` parameter should end with the suffix attribute from the `slapd.conf` file.

```
Wolf:~ alex$ ldapsearch -D "cn=ldapadmin,o=j2anywhere,c=gb" -w secret -x -b "ou=
people, o=j2anywhere, c=gb"
# extended LDIF
#
# LDAPv3
# base <ou=people, o=j2anywhere, c=gb> with scope sub
# filter: (objectclass=*)
# requesting: ALL
#
# people, j2anywhere, gb
dn: ou=people,o=j2anywhere,c=gb
objectClass: organizationalUnit
ou: people

# Alexander Hartner, people, j2anywhere, gb
dn: cn=Alexander Hartner,ou=people,o=j2anywhere,c=gb
displayName: Alexander Hartner
...

```

```
mail: alex@j2anywhere.com

# search result
search: 2
result: 0 Success

# numResponses: 3
# numEntries: 2
Wolf:~ alex$
```

Now you can configure Address Book 2 / 4 / X LDAP and transfer contacts. To connect to the directory as configured in the examples use

**Server** : *Hostname or IP address of the OpenLDAP Server*

**User / BindDN** : *cn=ldapadmin,o=j2anywhere,c=gb*

**Password** : *secret*

**Context / Search Base** : *ou=people, o=j2anywhere, c=gb*

The bind DN should match the rootdn in slapd.conf, while the password must be the same password used when you generated the hashed password using ldappasswd. The context should end with the suffix specified.

### **Recommended Reading**

I suggest you have a look at the **man pages** for the commands used to obtain information on the various switches.

# Startup Items

## Instructions for Tiger

### Option 1 - Launch Daemon

Available on the website you will find the openldap launch daemon configuration (org.openldap.slapd.xml.tar.gz). All you need to do is to download this file and extract its contents. Copy the configuration file (org.openldap.slapd.xml) into the /Library/LaunchDaemons directory. Once the file has been installed the service can be started and stopped using the following commands.

**Please stop the service started earlier in a terminal window by pressing CTRL+C after focusing the terminal window. A prompt should confirm the LDAP is stopped.**

To Start:

```
sudo launchctl load /Library/LaunchDaemons/org.openldap.slapd.xml
```

To Stop:

```
sudo launchctl unload /Library/LaunchDaemons/org.openldap.slapd.xml
```

For more information on using launchctl please take a look at the man page

To verify that the service is configured use as root:

```
Wolf:~ alex$ sudo launchctl list
Password:
com.apple.KernelEventAgent
com.apple.dashboard.advisory.fetch
com.apple.dnbsobserverd
com.apple.mDNSResponder
com.apple.nibindd
com.apple.periodic-daily
com.apple.periodic-monthly
com.apple.periodic-weekly
com.apple.portmap
com.apple.syslogd
com.vix.cron
org.postfix.master
org.xinetd.xinetd
com.apple.cups-lpd
com.openssh.sshd
org.openldap.slapd
Wolf:~ alex$
```

### Option 2 - Startup Items (not recommended)

Unfortunately the StartupItems for LDAP have been removed from Tiger and replaced with Launch Daemons. With the Address Book4LDAP application you also got the startup items for LDAP. (Separate download) You simply have to copy them to **/System/Library/StartupItems**. The only step left for you is to change the permissions for the new StartupItems to match those of the other startup items.

```
chown -R root:wheel LDAP
```

## Instructions for Panther

To startup LDAP automatically during startup add the following line to /etc/hostconfig.

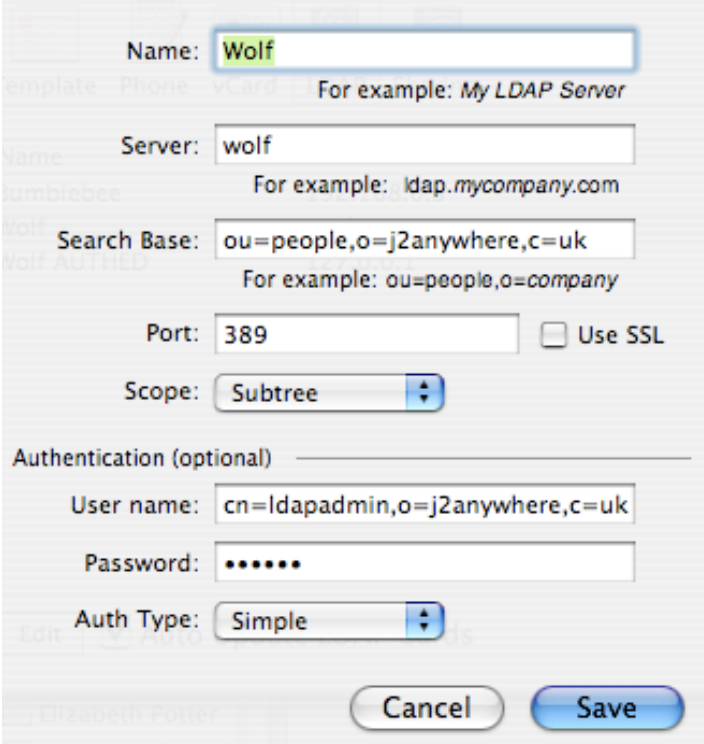
```
LDAPSERVER=-YES-
```

*Optionally, if you are using a LDAP configuration other than /etc/openldap/slapd.conf you have to modify /System/Library/StartupItems/LDAP/LDAP to include the modified configuration file or replace the default configuration file (/etc/openldap/slapd.conf) with the custom configuration file (/etc/openldap/Address Book\_slapd.conf).*

Now LDAP should start automatically after reboot. Try the ldapsearch command used earlier to verify and test your directory is operational. The next section covers various tools which interact with your LDAP directory.

## Address Book configuration

Apple's Address Book can be configured to use an LDAP directory.



The screenshot shows the configuration dialog for an LDAP directory in Apple's Address Book. The fields are as follows:

- Name:  (Example: My LDAP Server)
- Server:  (Example: ldap.mycompany.com)
- Search Base:  (Example: ou=people,o=company)
- Port:   Use SSL
- Scope:
- Authentication (optional):
  - User name:
  - Password:
  - Auth Type:

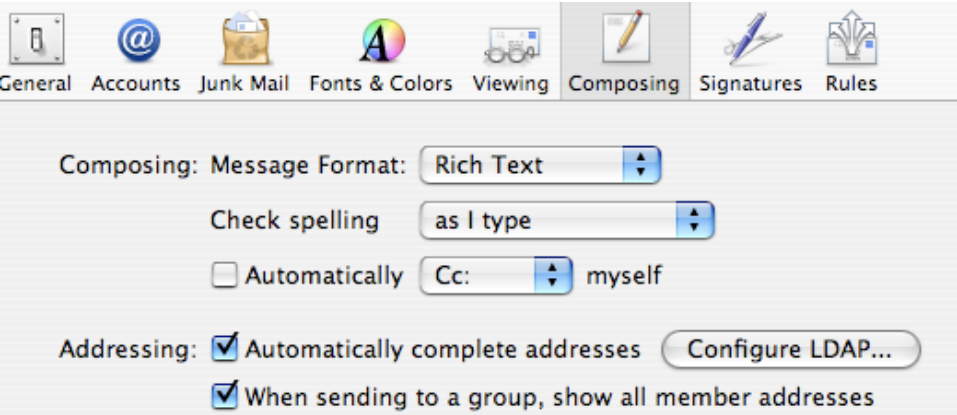
Buttons:

Once the Address Book is configured to search LDAP, search the directory via the search field. The application does not allow browsing of contacts in the directory. If you wish to browse the directory you can use Address Book4LDAP.



## Mail configuration

Similarly Mail can be configured to search LDAP for email addresses.



The screenshot shows the configuration dialog for Mail, specifically the Composing and Addressing sections. The settings are as follows:

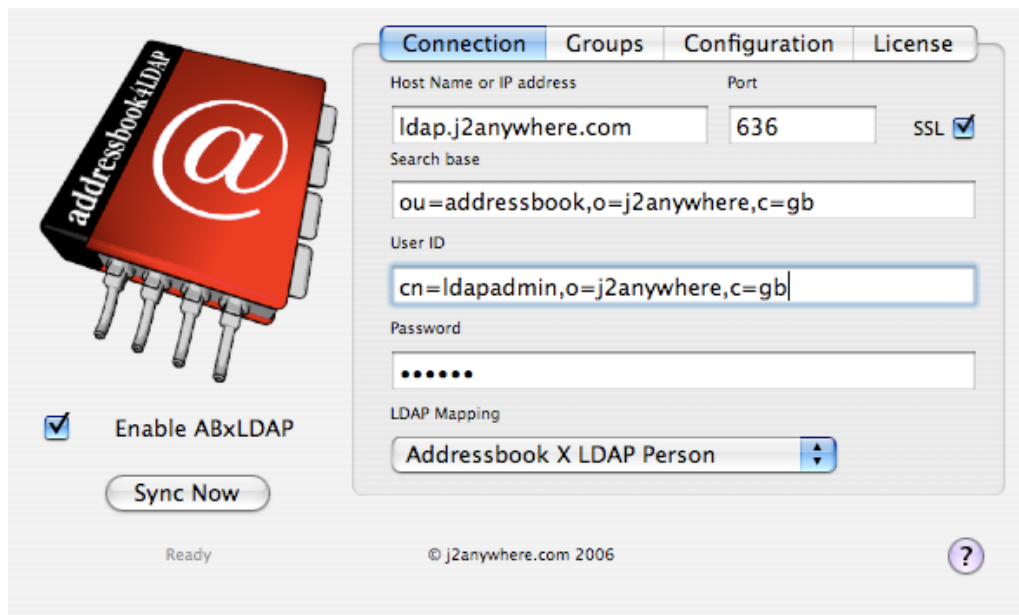
- Composing: Message Format:
- Check spelling:
- Automatically Cc:
- Addressing:  Automatically complete addresses
- When sending to a group, show all member addresses

# Address Book X LDAP

Address Book X LDAP consist of several parts and includes a Preference Panel (ABxLDAP Prefs), a LDAP viewer (Address Book 4 LDAP v2) and a synchronisation tool (ABxLDAPTool).

## ABxLDAP Preference Panel

The connection to the directory can be configured using ABxLDAP Preferences which is accessible in System Preferences. Once ABxLDAP is enabled, changes made to the local Address Book are automatically transferred to the LDAP directory in the background.

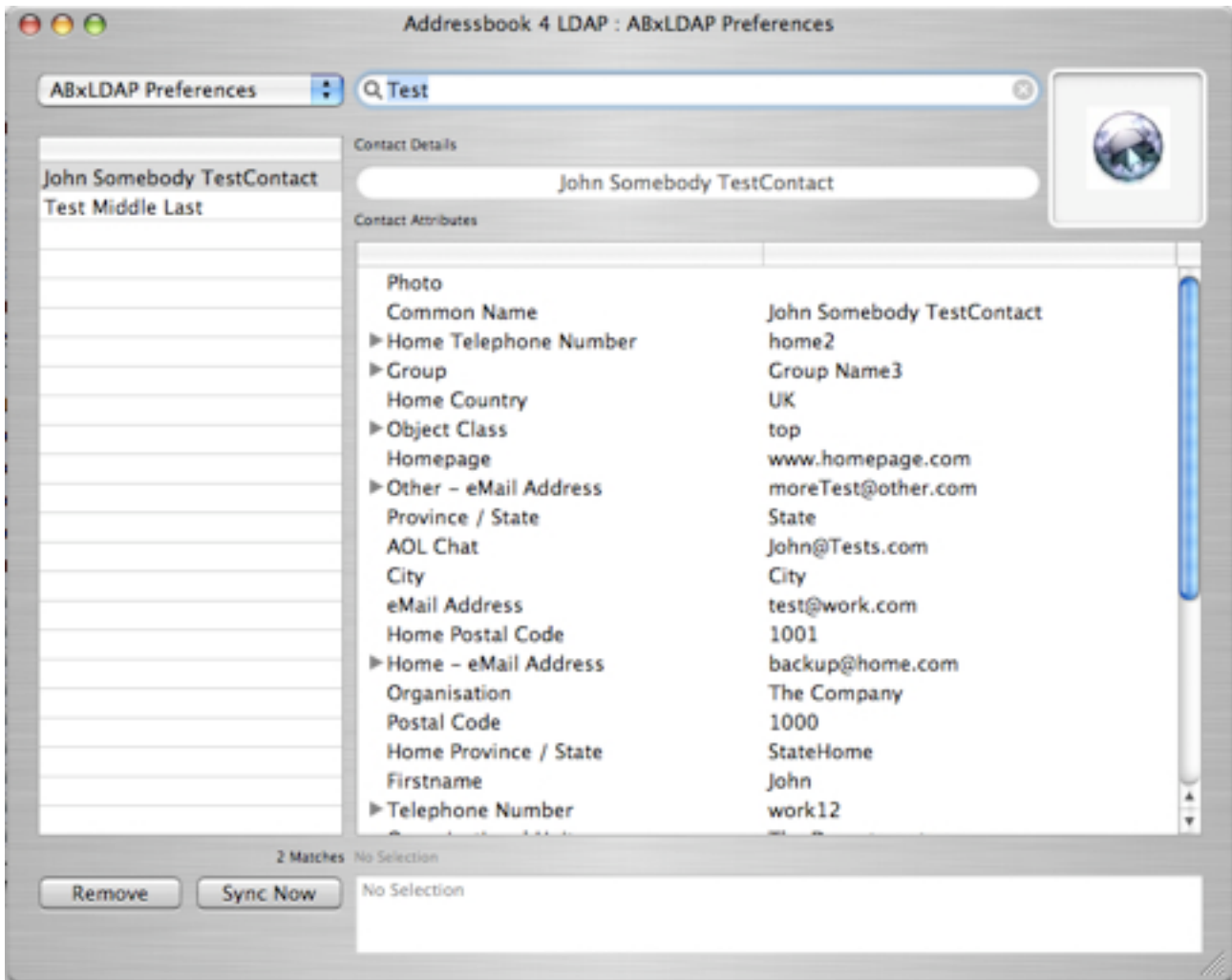


Additional configuration options are available and can be accessed in the application folders. The **/Library /PreferencePanes /ABxLDAP.prefPane /Contents /Resources** folder contains **ABxLDAPTool**, which performs the background synchronisation task, but can also be executed from a Terminal / Command line environment.

The schema mapping configuration resides in the **/Library/Application Support/ ABxLDAP** folder. Each mapping is associated with a mapping file, so AddressBook X LDAP Person is configured in **com.j2anywhere.ABxLDAP.LDAPMAP.plist**, for example. Any fields not contained within the mapping file are ignored. Please not that any changes made to the configuration are overwritten if you re-install the application. Please submit changes made for inclusion in future version.

## Address Book 4 LDAP (v2)

Address book 4 LDAP (Version 2) provides access to all attributes of the LDAP directory. As the standard Address book only provides access to a limited subset of attributes, this tool can be used to access the complete contact details. It also supports multi-values.



## Address Book 2 LDAP and Address Book 4 LDAP

Instructions for Address Book 2 / 4 LDAP are available online. As these projects are considered legacy they are excluded from this manual. Additional support for these projects is available and can be requested on the web site.

## Extending schema mappings

To add other attribute the schema mapping has to be configured. Here are the steps you need to follow :

1.) Identify the attributes which are not matched by looking at the output on the Console (Application / Utilities / Console) during a sync session. At the end of the sync session all mis-matched attributes are listed

```
ABxLDAPTool[4456] FAILED TO TRANSFORM ATTRIBUTE : countryCode:work
ABxLDAPTool[4456] FAILED TO TRANSFORM ATTRIBUTE : homeFax
ABxLDAPTool[4456] FAILED TO TRANSFORM ATTRIBUTE : displayAs
ABxLDAPTool[4456] FAILED TO TRANSFORM ATTRIBUTE : workim:jabber
ABxLDAPTool[4456] FAILED TO TRANSFORM ATTRIBUTE : countryCode:home
```

2.) Add a mapping for the identified attributes to the mapping file in

**/Library/Application Support/ABxLDAP/com.j2anywhere.ABxLDAP.LDAPMAP.plist**

This would usually mean adding a section like

```
<key>workim:jabber</key>
<string>homeimaim</string>
```

which maps the **workim:jabber** attribute to the **homeimaim** attribute in LDAP. If you don't want to use an existing ldap attribute you can also further extend the schema.

**If you make changes to the mapping, please be aware the future releases will overwrite your changes. If you like you can forward your changes for inclusion future versions.**

## Uninstalling

To remove ABxLDAP from your system delete the following folders:

### **Attribute Mapping configurations:**

/ Library / Application Support / ABxLDAP

### **ABxLDAP Shared components:**

/ Library / Frameworks / ABxLDAP.framework

### **ABxLDAP Shared components:**

/ Library / PreferencePanels / ABxLDAP.prefPane

### **Addressbook4LDAP Tool:**

/ Applications / AddressBook4LDAP

### **Application Preferences :**

~/ Library / Preferences / com.j2anywhere.abxldap.plist

~/ Library / Preferences / com.j2anywhere.AddressBook4LDAP.plist



# Thunderbird Address Book

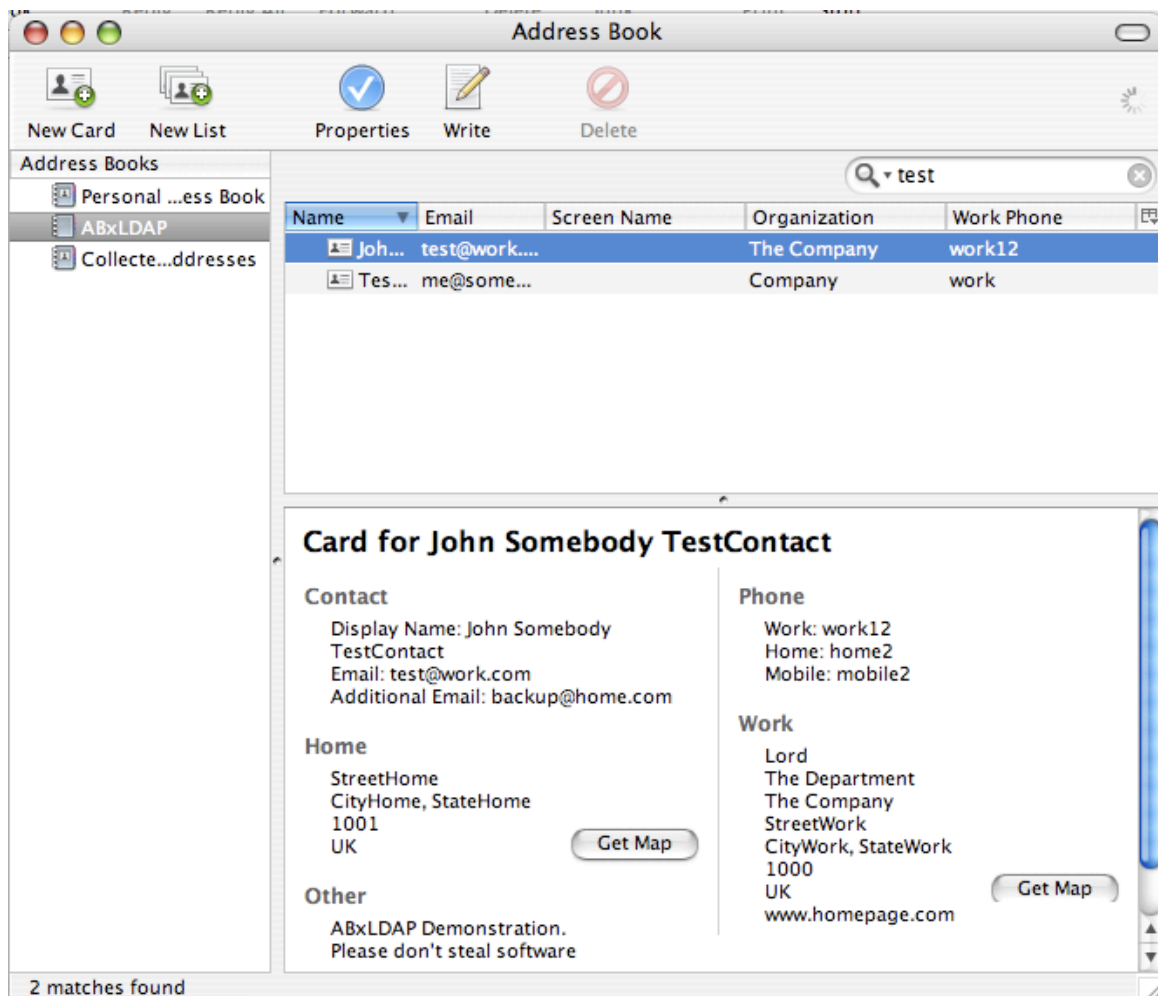
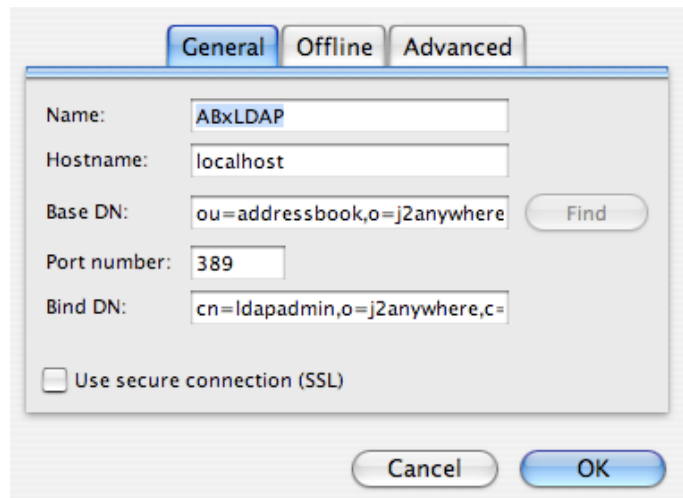
The Address Book included with Thunderbird has excellent support for LDAP integration. Not only does it offer SSL support, but it also offers a schema abstraction layer. This abstraction layer allows the directory to interpret different LDAP schema.

## Connection Configuration

The usual parameters are required to get Thunderbird to access the LDAP directory. Changes to the parameters, might require a restart of the entire application before taking effect.

## Schema Configuration

The Thunderbird Address Book uses its own schema mapping. It can be configured to make use of an alternate mapping, such as the Address Book X LDAP Person for example. The following addition to the prefs.js file in your users home directory are sufficient for the Address Book to correctly interpret the Address Book X LDAP Person schema.



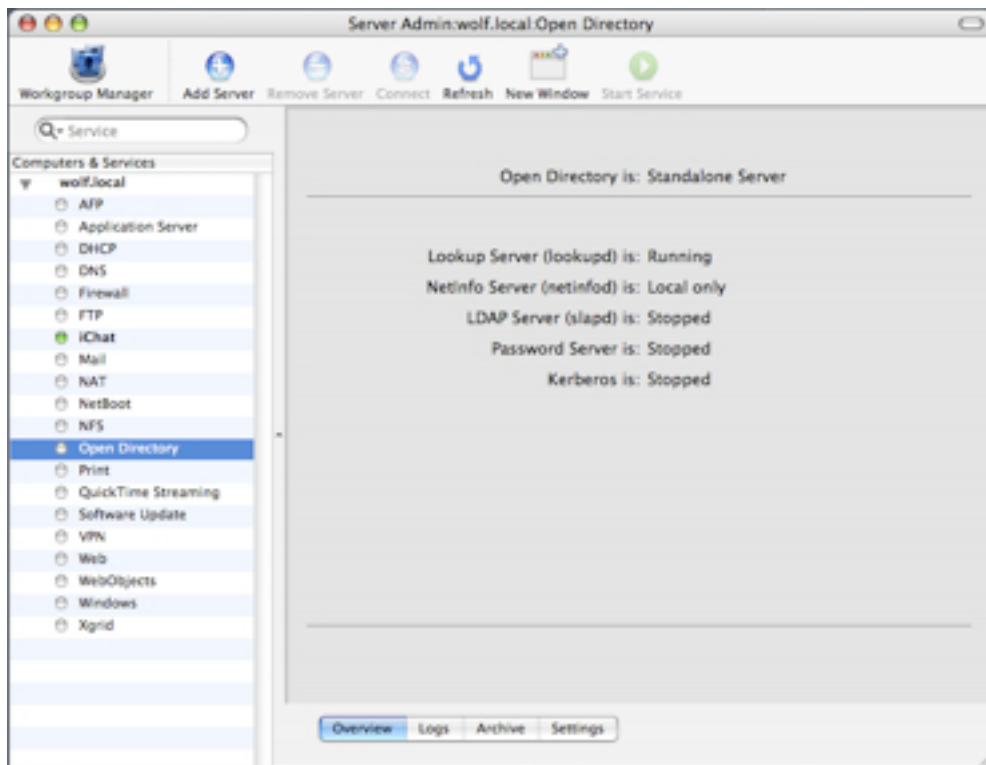
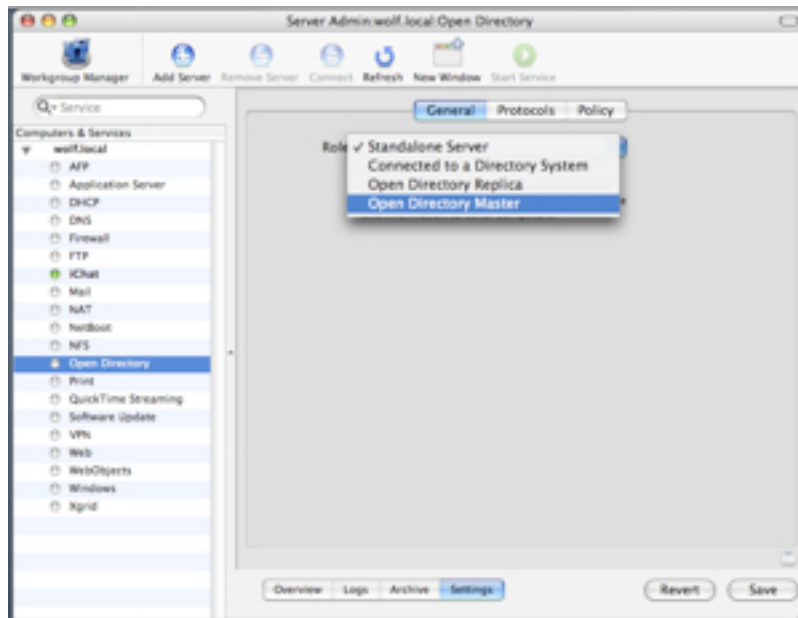
Even though this provides a lot of flexibility, and set it apart from most other tools, it does not support multi-value fields. So should one contact have two work phone numbers, both are not accessible, as only the first is shown.

prefs.js in users home directory /Library/Thunderbird/Profiles/[RANDOM]/prefs.js

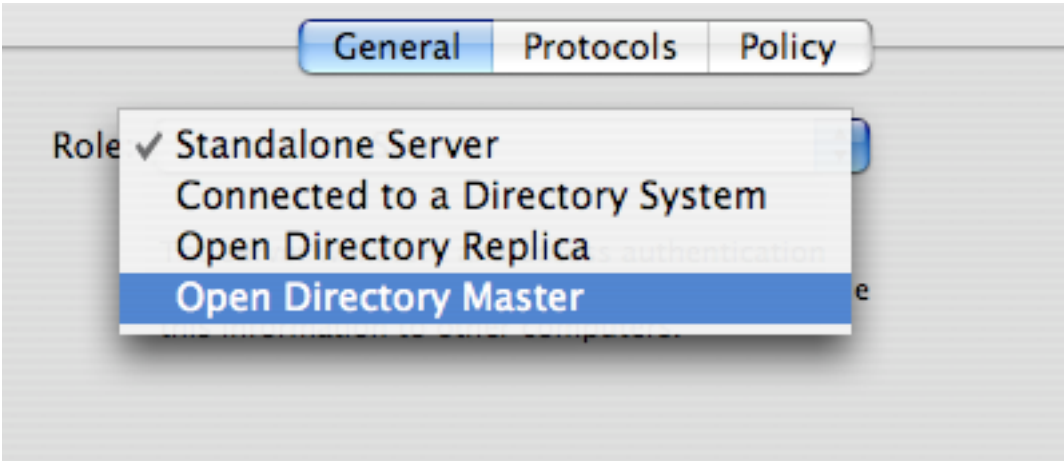
```
user_pref("ldap_2.servers.default.attrmap.CellularNumber", "mobile");
user_pref("ldap_2.servers.default.attrmap.DisplayName", "displayName");
user_pref("ldap_2.servers.default.attrmap.HomeAddress", "homeStreet");
user_pref("ldap_2.servers.default.attrmap.HomeCity", "homeCity");
user_pref("ldap_2.servers.default.attrmap.HomeCountry", "homeCountry");
user_pref("ldap_2.servers.default.attrmap.HomeState", "homeState");
user_pref("ldap_2.servers.default.attrmap.HomeZipCode", "homePostalCode");
user_pref("ldap_2.servers.default.attrmap.SecondEmail", "homeMail");
user_pref("ldap_2.servers.default.attrmap.WebPage1", "labeledURI");
user_pref("ldap_2.servers.default.attrmap.WorkAddress", "street");
user_pref("ldap_2.servers.default.attrmap.WorkCity", "l");
user_pref("ldap_2.servers.default.attrmap.WorkCountry", "c");
user_pref("ldap_2.servers.default.attrmap.WorkState", "st");
user_pref("ldap_2.servers.default.attrmap.WorkZipCode", "postalCode");
user_pref("ldap_2.servers.default.attrmap._AimScreenName", "homeimaim");
```

# OS X Server Installation

OpenDirectory included with OS X server makes the installation a little easier as some aspects can be configured using the server tools. A fresh installation of OS X server only has a Standalone Server configured. A Standalone Server only supports lookupd and netinfo, but not ldap.



To enable LDAP services change the Open Directory Server type to **Open Directory Master** under the General Tab.



The configuration of an Open Directory master domain requires a username (default : diradmin) as well as a password. The password selection is important it this is the password we will be using later on when we connect ABxLDAP. The search base is optional. Once completed **create** the service.

**Create a new Open Directory master domain**

Creating a new Open Directory master domain requires you to create a new administrator account for that domain. This account needs to have a unique name, short name and user ID.

**New Account** Role: Open Directory Master

Name:

Short Name:  User ID:

Password:

Verify:

**Domain Info**

Kerberos Realm:

Search Base:

Search base is optional.

After the services has been created Open Directory services show show up like this with-  
ing the Server Admin application.



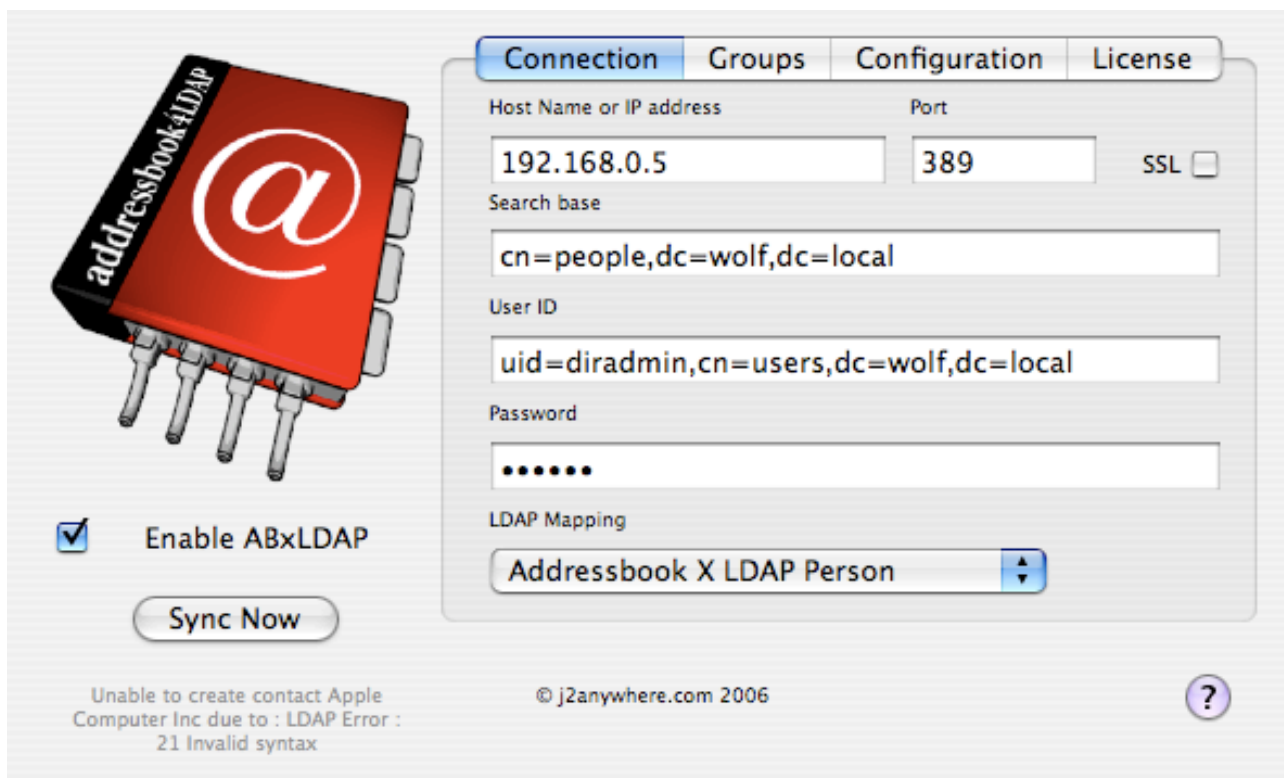
At this stage the Addressbook X LDAP Person Schema extension should be configured. This requires manual modifying the `/etc/ldap/slapd.conf` file. The process is the same for OS X and OS X server. Complete details are included earlier on in this manual.

After the schema extension has been included the server needs to be restarted.

If you know of a different way to restart OpenDirectory without a complete server restart please let me know

The clients can now be configured to access the LDAP directory. The username generated earlier can now be used to access the directory. Be careful to specify the fully qualified name as in `uid=diradmin,cn=users,dc=wolf,dc=local` rather than just `diradmin`.

On OS X Server the directory comes with a number of existing contexts. I suggest you use the `cn=people` domain for contact information.



The complete list of generated contexts can be accessed by querying the directory as follows :

```
wolf:~ alex$ ldapsearch -x -b dc=wolf,dc=local dn
# extended LDIF
#
# LDAPv3
# base <dc=wolf,dc=local> with scope sub
# filter: (objectclass=*)
# requesting: dn
#
```

```
# wolf.local
dn: dc=wolf,dc=local

# users, wolf.local
dn: cn=users,dc=wolf,dc=local

...
# people, wolf.local
dn: cn=people,dc=wolf,dc=local
...

# search result
search: 2
result: 0 Success

# numResponses: 32
# numEntries: 31
```

## Appendix A : SSL Configuration

The following section covers the installation and configuration of running LDAP over SSL. It is not required for LDAP to function but it is recommended to be used if you want to share your directory over the internet or with untrustworthy parties.

### Obtaining a SSL certificate

You can obtain a SSL server certificate from various sources such as Verisign and Thawte on the internet. Alternatively you could save your money ( or donate it to a software developer ;- ) and generate your own. The certificate you generate yourself will have the same strength as the ones available online, but you need to do a little more work. Also not everyone will trust your certificate automatically and would require your public certificate. The following paragraphs explain how to configure a certificate authority and how to generate a certificate. If you are going to get your certificate online, or already own one please continue with the configuration step.

### Creating a certificate authority

Start out by creating a new directory

```
mkdir j2anywhereCA
```

The following command will start the process

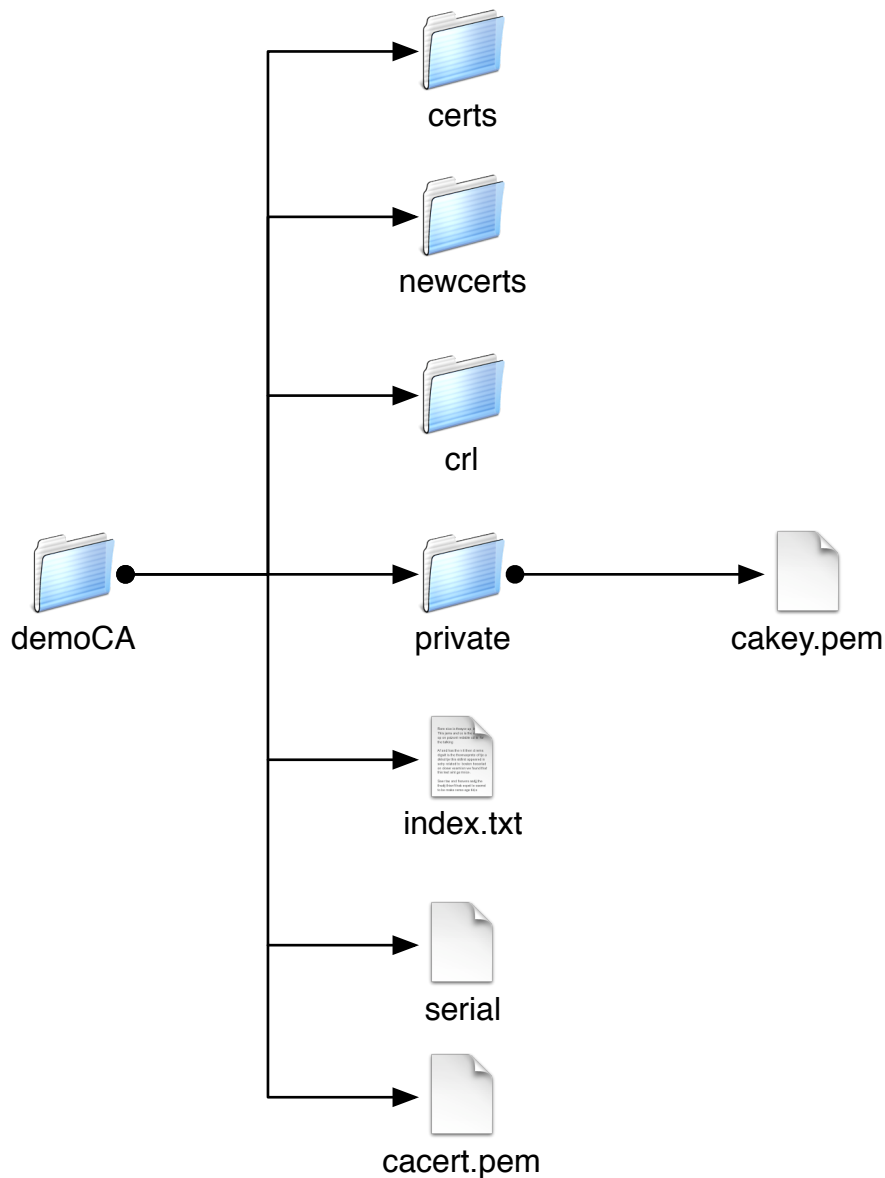
```
/System/Library/OpenSSL/misc/CA.pl -newca
```

A series of prompts will take you through the process

```
wolf:~/j2anywhereCA alex$ /System/Library/OpenSSL/misc/CA.pl -newca
CA certificate filename (or enter to create)
[ENTER]
Making CA certificate ...
Generating a 1024 bit RSA private key
.....++++++
...++++++
writing new private key to './demoCA/private/cakey.pem'
Enter PEM pass phrase:[your password]
Verifying - Enter PEM pass phrase:[your password]
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:GB
State or Province Name (full name) [Some-State]:Oxon
Locality Name (eg, city) []:Wallingford
Organization Name (eg, company) [Internet Widgits Pty Ltd]:j2anywhere.com
Organizational Unit Name (eg, section) []:Certificate Authority
Common Name (eg, YOUR name) []:j2anywhere.com
Email Address []:ca@j2anywhere.com
wolf:~/j2anywhereCA alex$
```

In the example the PEM password is set to password. The result of this command is a new directory named **demoCA**. The file **cacerts.pem** contains a self-signed certificate (including public key). The private key resides in the **private/cakey.pem** file.





## Creating a Certificate

To create a certificate with openssl the following command can be used.

```

wolf:~/j2anywhereCA alex$ /System/Library/OpenSSL/misc/CA.pl -newreq
Generating a 1024 bit RSA private key
.....+++++
.....+++++
writing new private key to 'newreq.pem'
Enter PEM pass phrase:[your password]
Verifying - Enter PEM pass phrase:[your password]
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
  
```

```
Country Name (2 letter code) [AU]:GB
State or Province Name (full name) [Some-State]:Oxon
Locality Name (eg, city) []:Wallingford
Organization Name (eg, company) [Internet Widgits Pty Ltd]:j2anywhere.com
Organizational Unit Name (eg, section) []:LDAP Directory
Common Name (eg, YOUR name) []:j2anywhere.com
Email Address []:ldap@j2anywhere.com
```

```
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []: [your password]
An optional company name []:j2anywhere.com
Request (and private key) is in newreq.pem
wolf:~/j2anywhereCA alex$
```

Be careful to use your server DNS name in the **Common Name** parameter, otherwise it will not work as the certificate will not match your server. This command will generate a file newreq.pem containing a certificate and a private key. To sign the generated certificate signing request CSR the following command is used :

```
wolf:~/j2anywhereCA alex$ /System/Library/OpenSSL/misc/CA.pl -sign
Using configuration from /System/Library/OpenSSL/openssl.cnf
Enter pass phrase for ./demoCA/private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
    Serial Number:
        89:3b:a1:e3:9d:7d:f0:80
    Validity
        Not Before: Mar 17 19:54:41 2006 GMT
        Not After : Mar 17 19:54:41 2007 GMT
    Subject:
        countryName           = GB
        stateOrProvinceName   = Oxon
        localityName          = Wallingford
        organizationName       = j2anywhere.com
        organizationalUnitName = LDAP Directory
        commonName             = j2anywhere.com
        emailAddress           = ldap@j2anywhere.com
    X509v3 extensions:
        X509v3 Basic Constraints:
            CA:FALSE
        Netscape Comment:
            OpenSSL Generated Certificate
        X509v3 Subject Key Identifier:
            B4:19:8E:D9:53:3F:F7:8B:EB:82:C4:BF:E0:23:B7:B9:3B:CF:E5:62
        X509v3 Authority Key Identifier:

keyid:DE:C2:57:28:FC:28:2F:2E:BD:C3:8D:89:63:1E:44:21:72:07:93:1F

DirName:/C=GB/ST=Oxon/L=Wallingford/O=j2anywhere.com/OU=Certificate
Authority/CN=j2anywhere.com/emailAddress=ca@j2anywhere.com
        serial:89:3B:A1:E3:9D:7D:F0:7F

Certificate is to be certified until Mar 17 19:54:41 2007 GMT (365 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
```

```
Write out database with 1 new entries
Data Base Updated
Signed certificate is in newcert.pem
```

This will create **newcert.pem** which contains a signed certificate, signed by your own certificate authority.

To extract the private key from the request use the following.

```
wolf:~/j2anywhereCA alex$ openssl rsa < newkey.pem > server_key.pem
Enter pass phrase:
writing RSA key
wolf:~/j2anywhereCA alex$
```

Often it is helpful to rename the created file names as follows:

```
mv newcert.pem server_cert.pem
mv newreq.pem server_req.pem
```

### Certificate files

You should now end up with the following files

- **server\_cert.pem** is the signed certificate
- **server\_req.pem** is the certificate signing request
- **server\_key.pem** is the private key
- **cacert.pem** is the public certificate of the certificate authority
- 

### Configuration of OpenLDAP directory

Append the following lines at the end of the slapd.conf file. Also copy the generated certificates into the appropriate directory.

```
TLSCipherSuite HIGH:MEDIUM:+SSLv2
TLSCACertificateFile /etc/openldap/cacert.pem
TLSCertificateFile /etc/openldap/server_cert.pem
TLSCertificateKeyFile /etc/openldap/server_key.pem
```

**WARNING : It is required that the certificate authority's certificate is named cacert.pem and is stored in the /etc/openldap folder.**

Once ldap has been restarted the connection can be tested with the following command.

```
openssl s_client -connect 66.116.103.223:636
```

After restarting LDAP you should be able to access the directory via

```
ldapsearch -D "... " -w ... -x -H ldaps://127.0.0.1:636 -b ".. "
```

If you experience a problem try starting LDAP in debug mode as follows :

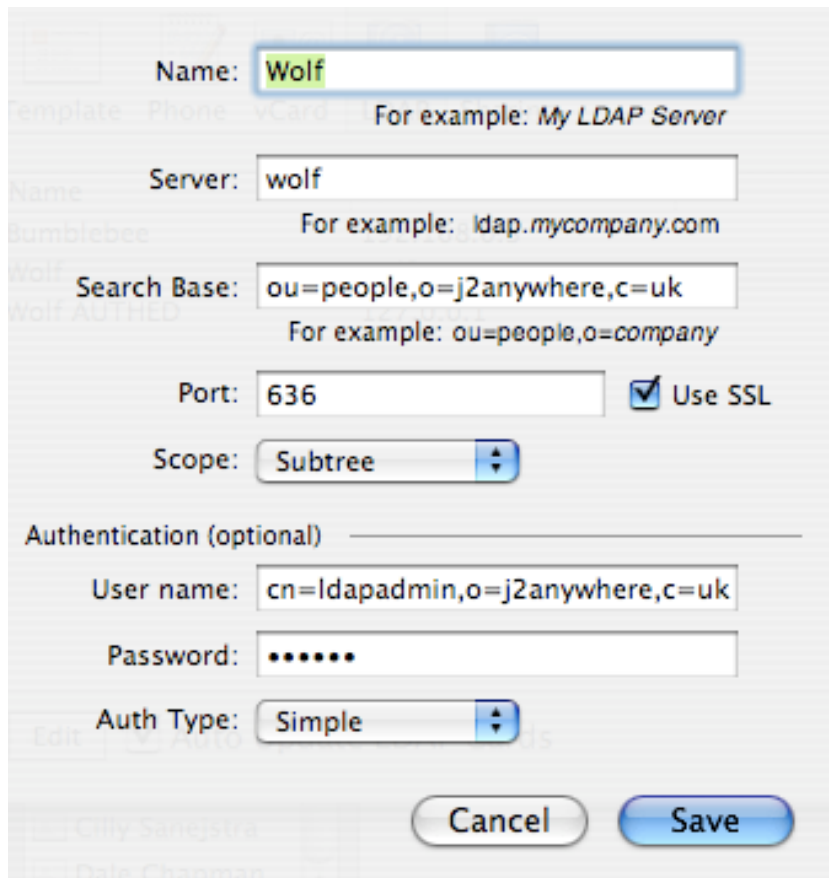
```
/usr/libexec/slapd -h "ldap:/// ldaps:///" -d 255 -f /etc/openldap/slapd.conf
```

The added **-h** parameter will allow access to both port 389 LDAP and 636 LDAPS. To test the server use the following command.

```
ldapsearch -D "... " -w ... -x -H ldaps://wolf:636 -Z -b "... "
```

## Client Configuration

Address Book4 / X LDAP have support for SSL connection. Even though this option is available in the current Address Book preferences there seem to be issues with using SSL from within Address Book. In the future this might change. For now you need to configure a new LDAP connection. The server should match server DNS name which should also match the the CN provided during the certificate creation. The ou should be the same as the suffix parameter specified in slapd.conf. Depending on the options either us port 389 or 636 (SSL). You don't need it for Apple's Address Book necessarily, but Address Book 4 LDAP requires the Authentication information. The user name should be the same the as rootdn in slapd.conf, and similarly the password should match the password provided to the slappasswd command. All that is left now is to start Address Book 4 LDAP. If you experience any problems please use the feedback page on the website.



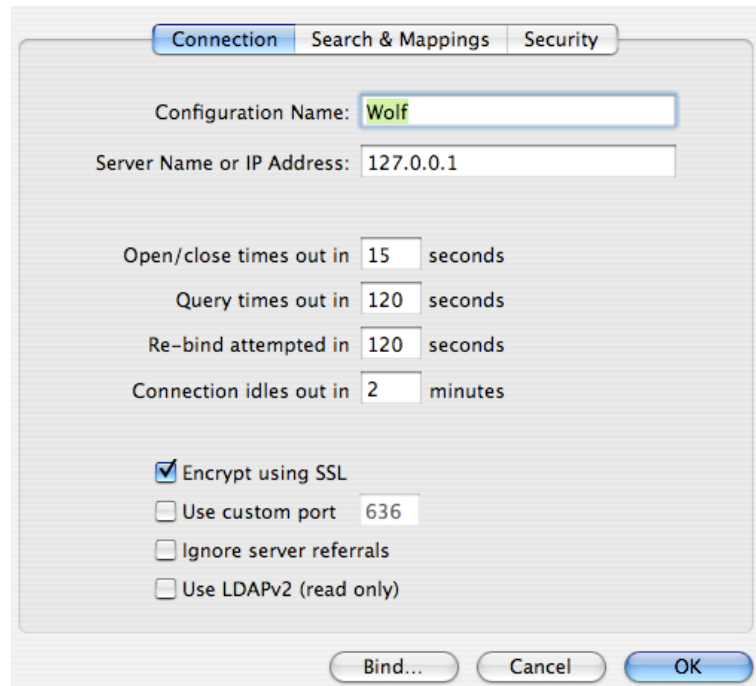
The image shows a dialog box for configuring an LDAP connection. The fields are as follows:

- Name:** Wolf (highlighted in green)
- Server:** wolf
- Search Base:** ou=people,o=j2anywhere,c=uk
- Port:** 636
- Use SSL:**
- Scope:** Subtree
- Authentication (optional):**
  - User name:** cn=ldapadmin,o=j2anywhere,c=uk
  - Password:** ••••••
  - Auth Type:** Simple

Buttons: Cancel, Save

## Warning – Address Book & SSL Problem !!!

Apple's Address Book does not seem to support SSL connection directly. You need to configure Directory Services to use SSL.



## Appendix B : Client Authentication

[Explanation on configuring individual users access to the LDAP directory to go here]

```
# Match the DN to the userid using a regular expression
access to dn.regex="^.*,cn=( [^, ]+ ),ou=users,o=j2anywhere,c=gb"
    by dn.exact,expand="cn=$1, ou=users,o=j2anywhere,c=gb"      read
    by *                                                         none

# Allow anybody to authenticate
access to *
    by anonymous auth
```

```
cn=user1,ou=users,o=j2anywhere,c=gb
cn=user2,ou=users,o=j2anywhere,c=gb
cn=user3,ou=users,o=j2anywhere,c=gb
```

and their own address book respectively

```
ou=addressbook,cn=user1,ou=users,o=j2anywhere,c=gb
ou=addressbook,cn=user2,ou=users,o=j2anywhere,c=gb
ou=addressbook,cn=user3,ou=users,o=j2anywhere,c=gb
```

## **Feedback**

Please include the relevant sections of the log file as well as any output from the console in any feedback. Either use the web-site : <http://j2anywhere.com> or e-mail [support@j2anywhere.com](mailto:support@j2anywhere.com)