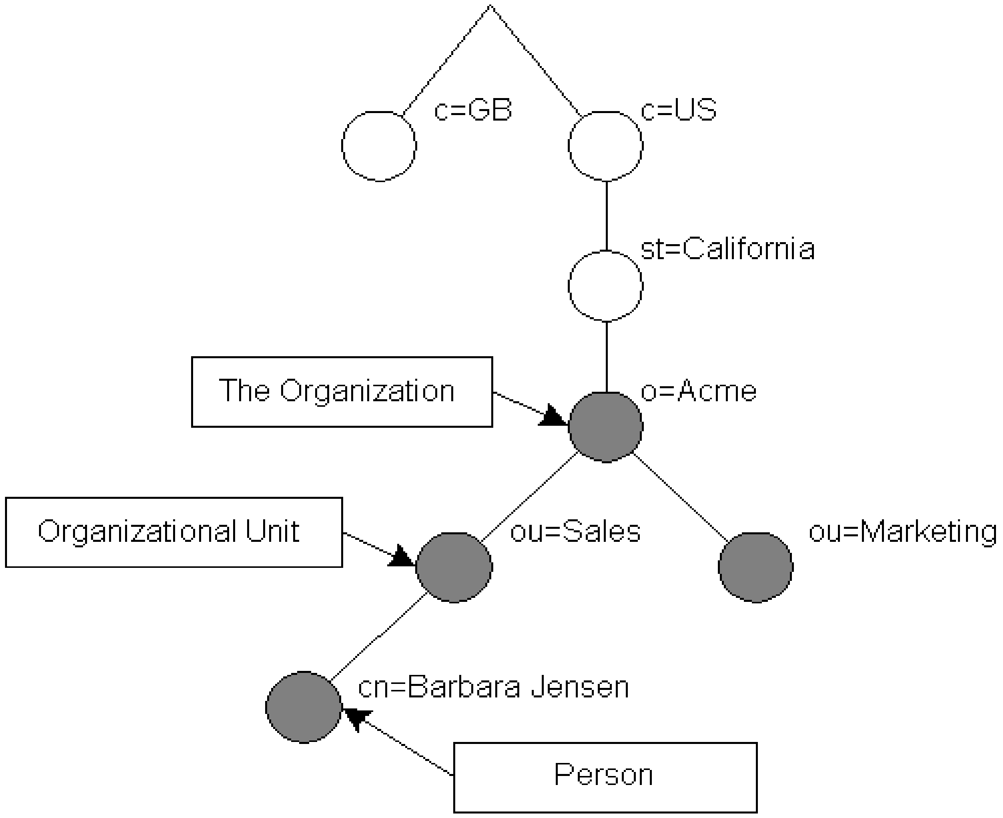
What is LDAP?

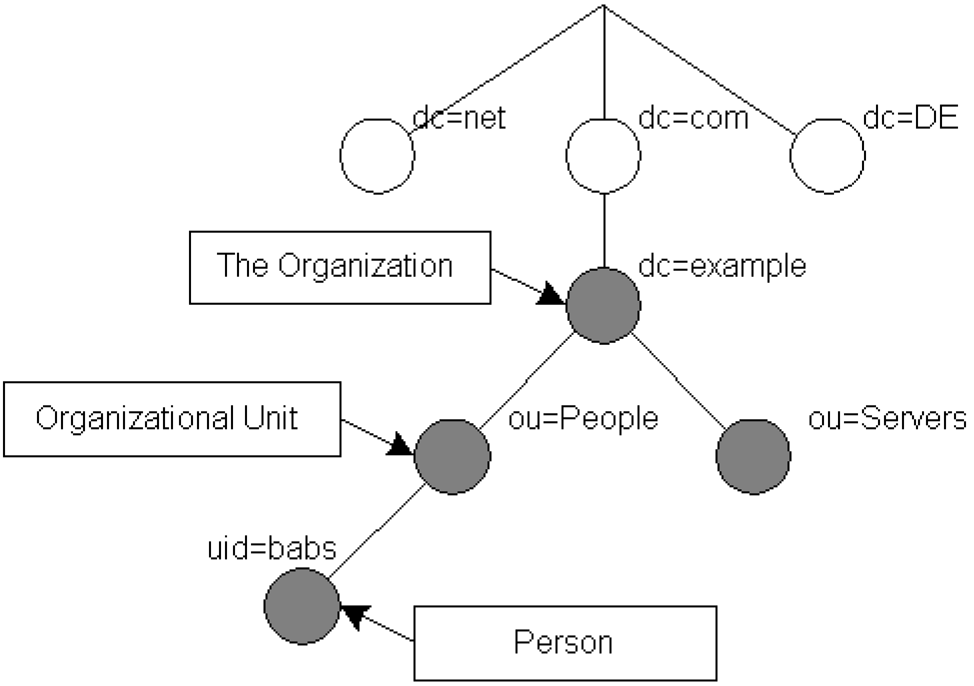
LDAP stands for Lightweight Directory Access Protocol. As the name suggests, it is a lightweight protocol for accessing directory services, specifically X.500-based directory services. LDAP runs over TCP/IP or other connection oriented transfer services. The nitty-gritty details of LDAP are defined in RFC2251 "The Lightweight Directory Access Protocol (v3)" and other documents comprising the technical specification RFC3377.

What kind of information can be stored in the directory? The LDAP information model is based on *entries*.An entry is a collection of attributes that has a globally-unique Distinguished Name (DN). The DN is used to refer to the entry unambiguously.

```
DN:relativeDomainName=domain1,dc=nic,dc=cctld
objectClass:dNSZone
objectClass:zonePerson
relativeDomainName:domain1
zoneName:cctld
dNSClass:IN
proprietaire:CLIENT1
dateacquis:20040604041800Z
validite:20060605164000Z
technical-contact: ALAIN AINA
technical-contact:AIT, bangkok
technical-contact:Tel:+78123455678-Email:aalain@trstech.net
admin-contact: John CRAIN
admin-contact:ICANN
admin-contact:Tel:+2282255555 - Email: john@icann.org
dNSTTL:7200
nSRecord: adjo.cafe.org.
nSRecord: ns.psg.com.
```

How is the information arranged? In LDAP, directory entries are arranged in a hierarchical tree-like structure. Traditionally, this structure reflected the geographic and/or organizational boundaries. The tree may also be arranged based upon Internet domain names. This naming approach is becoming increasing popular as it allows for directory services to be located using the *DNS*.


c=GB
c=US
st=California
The Organization
o=Acme
Organizational Unit
ou=Sales
ou=Marketing
cn=Barbara Jensen
Person


dc=net
dc=com
dc=DE
The Organization
dc=example
Organizational Unit
ou=People
ou=Servers
uid=babs
Person

In addition, LDAP allows you to control which attributes are required and allowed in an entry through the use of a special attribute called `objectClass`. The values of the `objectClass` attribute determine the *schema* rules the entry must obey.

```
objectclass ( 1.1.2.2.2 NAME 'myPerson'
      DESC 'my person'
      SUP inetOrgPerson
      MUST ( myUniqueName $ givenName )
      MAY myPhoto )

attributetype ( 1.1.2.1.2 NAME 'myPhoto'
       DESC 'a photo (application defined format)'
       SYNTAX 1.3.6.1.4.1.1466.115.121.1.40 SINGLE-VALUE )
```

How is the information referenced? An entry is referenced by its distinguished name, which is constructed by taking the name of the entry itself (called the Relative Distinguished Name or RDN) and concatenating the names of its ancestor entries. The full DN format is described in RFC2253, "Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names."

```
DN:relativeDomainName=domain1,dc=nic,dc=cctld
```

How is the information accessed? LDAP defines operations for interrogating and updating the directory. Operations are provided for adding and deleting an entry from the directory, changing an existing entry, and changing the name of an entry. Most of the time, though, LDAP is used to search for information in the directory. The LDAP search operation allows some portion of the directory to be searched for entries that match some criteria specified by a search filter. Information can be requested from each entry that matches the criteria.

How is the information protected from unauthorized access? Some directory services provide no protection, allowing anyone to see the information. LDAP provides a mechanism for a client to authenticate, or prove its identity to a directory server, paving the way for rich access control to protect the information the server contains. LDAP also supports privacy and integrity security services.

LDAP Model

LDAP models represent the services provided by a server, as seen by a client. They are abstract models that describe the various facets of an LDAP directory. RFC 2251 divides an LDAP directory into two components: the protocol model and the data model.

Information model

The information model provides the structures and data type necessary for building an LDAP directory tree. An entry is the basic unit in an LDAP directory. You can visualize an entry as either an interior or exterior node in the Directory Information Tree (DIT). An entry contains information about an instance of one or more objectClasses. These objectClasses have certain required or optional attributes. Attributes types have defined encoding and matching rules that govern such things as the type of data the attribute can hold and how to compare this data during a search.

Naming model

The naming model defines how entries and data in the DIT are uniquely referenced. Each entry has an attribute that is unique among all sibling of a single parent. This unique attribute is called the relative distinguished name (RDN). You can uniquely identify any entry within a directory by following the RDNs of all the entries in the path from the desired node to the root of the tree. This string created by combining RDNs to form a unique name is called the node's distinguished name (DN).

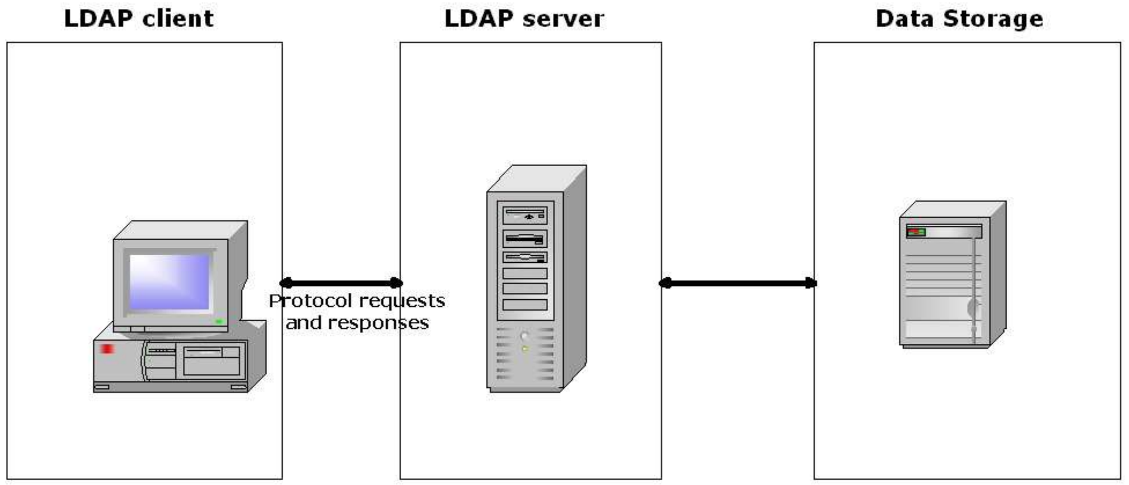
Functional model

The functional model is the LDAP protocol itself. This protocol provides the means for accessing the data in the directory tree. Access is implemented by authentication operations, query operations (searches and reads), and update operations (writes).

```
access to *
        by self write
         by anonymous auth
         by users read
```

Security model

The security model provides a mechanism for clients to prove their identity(authentication) and for the server to control an authenticated client's access to data(authorization). LDAPv3 provides several authentication methods not available in previous protocol versions. Some features, such as access control lists, have not been standardized yet, leaving vendors to their own devices.



Relationship between an LDAP client, LDAP server, and data storage facility

LDAP directory service is based on a *client-server* model. One or more LDAP servers contain the data making up the directory information tree (DIT). The client connects to servers and asks it a question. The server responds with an answer and/or with a pointer to where the client can get additional information (typically, another LDAP server). No matter which LDAP server a client connects to, it sees the same view of the directory; a name presented to one LDAP server references the same entry it would at another LDAP server. This is an important feature of a global directory service, like LDAP.

How to install openldap

Berkelley DB Installation

Home page http://www.sleepycat.com

. Now we must get the source code of Berkeley DB on http://www.sleepycat.com

```
$ tar -xvzf db-4.1.25.NC.tar.gz
$cd /tmp/db-4.1.25.NC/dist
$./configure
... takes a while
$make
$su –
Password: <root password>
#cd /tmp/db-4.1.25.NC/dist/
# make install
```

Install openldap

Get the software

You can obtain a copy of the software by following the instructions on the OpenLDAP download page (http://www.openldap.org/software/download/). It is recommended that new users start with the latest *release*

```
$tar -xvzf openldap-version.tar.gz
$cd openldap-version
$./configure --prefix=/usr --exec-prefix=/usr --libexecdir=/usr/sbin --sbindir=/usr/sbin --bindir=/usr/sbin --libdir=/usr/lib --oldicludedir=/usr/include --localstatedir=/var/run --sysconfdir=/etc --enable-shared  --with-gnu-ld --enable-debug --with-tls  --with-threads --enable-crypt  --enable-cleartext --enable-slapd --enable-slurpd --enable-bdb --enable-local --enable-passwd --enable-static --enable-FEATURE --with-PACKAGE  --enable-syslog -enable-ldap --with-readline
$ make depend
$make
$ cd tests
$ make
$ su –
Password:  <root password>
#cd /tmp/openldap-version
#make install
```

Edit the configuration File

Use your favourite editor to edit the provided slapd.conf example(usually installed as /etc/openldap/slapd.conf) to contain BDB database definition of of the form:

```
database bdb
suffix "dc=<MY-DOMAIN>,dc=<COM>"
rootdn "cn=Manager,dc=<MY-DOMAIN>,dc=<COM>"
rootpw secret
directory /var/openldap-data
```

Be sure to replace <MY-DOMAIN> and <COM> with the appropriate domain components of your domain name. For example, nic.cctld, use:

```
database bdb
suffix "dc=nic,dc=cctld"
rootdn "cn=Manager,dc=nic,dc=cctld"
rootpw secret
directory /var/openldap-data
```

You should be sure to specify a directory where the index files should be created. You need to create this directory with appropriate permissions such that slapd can write to it.

```
#mkdir –p /var/openldap-data
#chmod –R  700 /var/openldap-data
```

Start SLAPD

You are now ready to start the stand-alone LDAP server, by running the command: slapd . *To check to see if the server is running and configured correctly, you can run a search against it with with ldapsearch.*

```
ldapsearch -x -b ‘’   -s base ‘(objectclass=*)’ namingContexts
```

Note the use of single quotes around command parameters to prevent special characters from being interpreted by the shell. This should return:

```
dn:
namingContexts: dc=nic,dc=cctld
```

Add initial entries to your directory

You can use ldapadd to add entries to your LDAP directory. Ldapadd expects input in LDIF form. We will do it in two steps:

- create an LDIF file
- run ldapadd

Use your favorite editor and create an LDIF file that contains:

```
dn:dc=nic,dc=cctld
objectClass:dcObject
```

```
      objectClass:organization
      o:CCTLD COMPANY
      dc:cctld


      dn:cn=Manager,dc=nic,dc=cctld
      objectClass:organizationalRole
      cn:Manager
```

Now you may run ldapadd to insert these entries into your directory.

```
            ldapadd -x -D "cn=Manager,dc=nic,dc=cctld" -W -f cctld.ldif
```

```
where  cctld.ldif is the file you create above
```

See if it works

Now we are ready to verify the added entries are in your directory. You can use any LDAP client to do this, but our example uses the ldapsearch tool.

```
ldapsearch  -x -b  ‘dc=nic,dc=cctld’ ‘(objectclass=*)’
```

This command will search and retrieve every entry in the database. You are now ready to add more entries using ldapadd or another LDAP client, experiment with various configuration options, backend arrangements, etc…

Bind and LDAP

Get bind-9.3.0 from **www.isc.org**
Get bind-sdb-ldap-1.0-beta and dnszone.schema **from www.venaas.no/ldap/bind-sdb**
Get localzone.schema from http://www.trstech.net/registry/localzone.schema

1-Maintain your ldap schema directory

```
      # cp /tmp/dnszone.schema             /etc/openldap/schema/
      #cp /tmp/localzone.schema            /etc/openldap/schema/
```

Be sure that root is the owner of this two files.

2-Update your slapd.conf file by adding the following lines:

```
include   /etc/openldap/schema/core.schema
include   /etc/openldap/schema/cosine.schema
include   /etc/openldap/schema/dnszone.schema
include   /etc/openldap/schema/localzone.schema
#
index  relativeDomainName,zoneName              pres,eq
index  nSRecord,aRecord,sOARecord,mXRecord      pres,eq
```

3-Installation

```
$tar -xvzf /tmp/bind-9.3.0.tar.gz
$tar -xvzf /tmp/bind-sdb-1.0.tar.gz
$cp ./bind-sdb-1.0/ldapdb.c   /tmp/bind-9.3.0/bin/named
$cp ./bind-sdb-1.0/ldapdb.h /tmp/bind-9.3.0/bin/named/include/
```

4-Edit with your favorite editor /tmp/bind-9.3.0/bin/named/Makefile.in anda dd the following lines :

```
DBDRIVER_OBJS = ldapdb.@O@
DBDRIVER_SRCS = ldapdb.c
DBDRIVER_INCLUDES = -I/usr/local/include
DBDRIVER_LIBS = -L/usr/local/lib -lldap –llber
```

5-Edit /tmp/bind-9.3.0/bin/named/main.c and add:

- the line # include <ldapdb.h> below # include "xxdb.h"
- the line ldapdb_init(); below xxdb_init();
- the line ldapdb_clear(); below xxdb_clear();

6-After making this changes, you are ready to build the LDAP-enabled named binary by executing :

```
        $cd /tmp/bind-9.3.0/
        $./configure  --prefix=/usr --exec-prefix=/usr --bindir=/usr/bin --sbindir=/usr/sbin --
libexecdir=/usr/libexec --sysconfdir=/etc --localstatedir=/var --libdir=/usr/lib --oldincludedir=/usr/include/
--enable-static --enable-shared --enable-fast-install --with-gnu-ld with-pic
        $make
        $su –
        Password: <root password>
        #cd /tmp/bind-9.3.0
        #make install
```

run named -c /etc/namedb/named.conf –u bind -g and look for error message

7- create a ldif file(cctld.ldif) with some zone data (Zone cctld and two subdomains domain1 and domain2)

```
dn:relativeDomainName=@,dc=nic,dc=cctld
objectClass:dNSZone
objectClass:zonePerson
relativeDomainName:@
zoneName:cctld
dNSClass:IN
proprietaire: Republic of
sOARecord: ns1.cctld.net.     Admin.mail.cctld.       2004060101  21600 3600
      604800      3600
technical-contact:Alain AINA
technical-contact: ISOC EDU
technical-contact: Email: aalain@trstech.net
admin-contact: ISOC
admin-contact: US
admin-contact:Tel:+1711112255555 Email:admin@mail.cctld
nSRecord:ns1.cctld.net.
nSRecord:ns2.isoc.org.
nSRecord:ns3.icann.org.
nSRecord:ns4.nsrc.org.
```

```
dn:relativeDomainName=domain1,dc=nic,dc=cctld
objectClass:dNSZone
objectClass:zonePerson
relativeDomainName:domain1
zoneName:cctld
dNSClass:IN
proprietaire:CLIENT1
dateacquis:20040604041800Z
validite:20060605164000Z
technical-contact: ALAIN AINA
technical-contact:AIT, bangkok
technical-contact:Tel:+78123455678-Email:aalain@trstech.net
admin-contact: John CRAIN
admin-contact:ICANN
admin-contact:Tel:+2282255555 - Email: john@icann.org
dNSTTL:7200
nSRecord: adjo.cafe.org.
nSRecord: ns.psg.com.

dn:relativeDomainName=domain2,dc=nic,dc=cctld
objectClass:dNSZone
objectClass:zonePerson
relativeDomainName:domain2
zoneName:cctld
dNSClass:IN
proprietaire:Client2
dateacquis:20040604041800Z
validite:20060605164000Z
technical-contact: samuel SODATONOU
technical-contact:Tel:+22822192245
admin-contact: Steve Huter
admin-contact:Tel:+1645662219235 - Email: sghuter@nsrc.org
dNSTTL:7200
nSRecord: localhost.cctld.
nSRecord: ns.ripe.net.
```

8-Add cctldzone.ldif file to your ldap database:

```
#ldapadd  -x  -D “cn=Manager,dc=nic,dc=cctld”  -W  -f  cctldzone.ldif
Password: <ldap Manager password>
```

9-Edit /etc/named.conf and insert the following lines

```
zone "cctld"  IN {
        type master ;
        database "ldap ldap://localhost/dc=nic,dc=cctld??sub? 172800";
};
```

After change your named.conf file, restart named and run dig to search for your cctld, domain1.cctld and domain2.cctld soa and ns records

```
dig @localhost cctld soa +norec
dig @localhost cctld ns +norec
dig @localhost domain1.cctld
dig @localhost. domain2.cctld
```

Install php-ldapadmin

NB: you need apache +php with ldap support

Get the source file of phpldapadmin-0.9.4b from http://phpldapadmin.sourceforge.net/download.php

```
	#mkdir –p /var/www/html/ldap
	# cd /var/www/html/ldap
	# tar  -xvzf /tmp/phpldapadmin-0.9.4b.tar.gz
	# mv phpldapadmin-0.9.4b  ./phpldapadmin
	#cd phpldapadmin
	#cp config.php.example  ./config.php
```

Edit config.php file and change the server name, the base, the binddn and bind password
Example

```
	$servers[$i]['host']  = 'ldap://localhost';
	$servers[$i]['base'] = 'dc=nic,dc=cctld';
	$servers[$i]['port'] = 389;
	$servers[$i]['auth_type'] = 'config';
	$servers[$i]['login_dn'] = 'cn=Manager,dc=nic,dc=cctld';
	$servers[$i]['login_pass'] = 'secret';
```

After changing your config.php file, you can connect with your browser to this address
http://ip_address/ldap/phpldapadmin

Other LDAP clients

LDAPBROWSER : http://www.iit.edu/~gawojar/ldap/
WEB2LDAP : http://freshmeat.net/projects/web2ldap/
GQ :http://biot.com/gq/

Reference: http://www.trstech.net/registry