OpenLDAP client tools

Query LDAP directories

START HERE

`ldapsearch -x -H ldap://`*`<host>`*` -b `*`<base>`
Anonymous search

`ldapwhoami -x -H `*`<uri>`*` -D `*`<binddn>`*` -W`
Who am I bound as?

`ldapsearch -x -H `*`<uri>`*` -D `*`<binddn>`*` -W -b `*`<base>`
Authenticated search

`ldapmodify -x -H `*`<uri>`*` -D `*`<binddn>`*` -W -f `*`<file>`
Apply an LDIF change

`ldapsearch -x -b `*`<base>`*` "(uid=`*`<name>`*`)"`
Filter entries

`ldapsearch -x -b `*`<base>`*` "(cn=*)" cn mail`
Pick attributes

`ldapsearch -LLL -x -b `*`<base>`*
Clean LDIF output

`ldapsearch -x -s base -b "" -H `*`<uri>`*` +`
Probe the server (rootDSE)

Connect -H uri, -Z StartTLS → `bind` → **Bind** -x -D -W or SASL → `search` → **Search** -b base -s scope filter → `return` → **Entries** attrs, LDIF output

One ldapsearch run

CONCEPTS

Always add -x unless you want SASL: without it the tools try SASL and often fail with confusing errors.

Directory data is a tree: a DN names an entry, the base picks the subtree, the scope picks how deep.

Put the filter in quotes; shell characters like ( ) and * would otherwise be eaten.

CONNECT AND BIND

`-H ldap://`*`<host>`*`[:`*`<port>`*`]`
Server URI (plain or StartTLS)

`-H ldaps://`*`<host>`*
TLS from the first byte

`-H ldapi:///`
Local socket (as root, with -Y EXTERNAL)

`-x`
Simple bind instead of SASL

`-D `*`<binddn>`*
Bind as this DN

`-W`
Prompt for the password

`-y `*`<file>`*
Read password from a file

CONNECT AND BIND (cont.)

`-w `*`<password>`*
Password on command line
Visible in the process list and shell history; prefer -W or -y.

`-ZZ`
StartTLS, fail if it cannot be used
-Z alone continues in cleartext if StartTLS fails.

`-Y EXTERNAL -H ldapi:///`
SASL EXTERNAL over the local socket

`-Y GSSAPI`
Kerberos login

`-o nettimeout=`*`<s>`*
Network timeout in seconds

SEARCHING

`-b `*`<base>`*
Search base DN

`-b ""`
Empty base: rootDSE with -s base

`-s base / one / sub / children`
Search scope (default sub)

`"(`*`<attr>`*`=`*`<value>`*`)"`
Equality filter

`"(`*`<attr>`*`=`*`<pre>`*`*)"`
Substring/wildcard match

SEARCHING (cont.)

`"(`*`<attr>`*`=*)"`
Attribute is present

`"(&(`*`<f1>`*`)(`*`<f2>`*`))"`
AND of filters

`"(|(`*`<f1>`*`)(`*`<f2>`*`))"`
OR of filters

`"(!(`*`<f>`*`))"`
Negate a filter

`"(`*`<attr>`*` ⩾ `*`<n>`*`)"`
Greater or equal (<= for less)

`-z `*`<n>`*
Max entries returned

`-l `*`<seconds>`*
Time limit for the search

`\28 \29 \2a \5c`
Escape ( ) * \ in values

`-f `*`<file>`*` "(uid=%s)"`
One search per input line

ATTRIBUTES AND OUTPUT

`... `*`<filter>`*` cn mail`
Return only listed attributes

`"*" "+"`
User plus operational attributes

`1.1`
Return DNs only

`-LLL`
Plain LDIF, no comments or version

ATTRIBUTES AND OUTPUT (cont.)

`-L`
LDIF v1 with comments

`-A`
Attribute names only, no values

`-S `*`<attr>`*
Sort entries client-side

`-o ldif_wrap=no`
Don't wrap long LDIF lines

`-t`
Write binary values to temp files

`-E pr=`*`<n>`*`/noprompt`
Paged results
Needed for directories that cap result size; combine with -z 0 or leave unset.

`-v`
Verbose diagnostics

CHANGE ENTRIES

`ldapadd -x -D `*`<binddn>`*` -W -f `*`<file>`*
Add entries from LDIF

`ldapmodify -x -D `*`<binddn>`*` -W -f `*`<file>`*
Apply LDIF changes

`ldapdelete -x -D `*`<binddn>`*` -W "`*`<dn>`*`"`
Delete an entry

CHANGE ENTRIES (cont.)

`ldappasswd -x -D `*`<binddn>`*` -W -S "`*`<dn>`*`"`
Set a password (prompts)

`ldapwhoami -x -D `*`<binddn>`*` -W`
Show the identity after bind

`ldapmodify -c -S `*`<skipfile>`*
Keep going, save failed records

`ldapmodify -a`
Treat input as adds

`ldapmodify -n -v`
Dry run with verbose output

LDIF change record

Line	Meaning
`dn: <dn>`	Entry to change
`changetype: modify`	Modify (also add, delete, modrdn)
`add: / replace: / delete: <attr>`	Operation on an attribute
`-`	Ends one operation block

Separate records in an LDIF file with a blank line; the lone - line closes each modify block.

RECIPES

Export a subtree

```
# dump a whole subtree to a file
ldapsearch -x -LLL -H ldaps://<host> -D <binddn> -W \
  -b "<base>" "(objectClass=*)" > backup.ldif
```

Discover the base DN

```
# find the naming contexts the server offers
ldapsearch -x -LLL -H ldap://<host> -s base -b "" \
  namingContexts supportedSASLMechanisms
```

Query over StartTLS

```
# members of a group, only mail addresses
ldapsearch -x -LLL -ZZ -H ldap://<host> -D <binddn> -W -b <base> \
  "(&(objectClass=person)(memberOf=<groupdn>))" mail
```

Edit one attribute

```
# change a mail address
cat > fix.ldif <<'EOF'
dn: <dn>
changetype: modify
replace: mail
mail: <new>
EOF
ldapmodify -x -D <binddn> -W -f fix.ldif
```