Testing LDAP extensions

This page gives some hints about how to test the LdapPlugin extension, and provides some examples about deploying OpenLDAP to perform the tests.

Prerequistes

The examples in this page assume that you are working with a Linux server (Debian), with OpenLDAP 2.2 or greater.
The `slapd` server should have been installed and you should also have access to the Ldap utils (which usually comes with a separate package) namely:

- server tools: `slapadd`, `slapcat`
- client tools: `ldapsearch`, `ldapadd`, `ldapmodify`, `ldapdelete`

All the commands are run using the superuser (root) account.

Create the directory config file

The following config file is somewhat more complex than it could be, as it uses ACL, etc.
However this is a good base to elaborate a more complex LDAP setup and … that's the file I use to test the extension

```
# BDB backend in this example
database        bdb

# Maximum entries returned in a search
sizelimit       100

# Log connections, operations, results
# Do not forget to reduce the debug level once everything is up and running !
loglevel        768

suffix          "dc=example,dc=org"
rootdn          "uid=root,dc=example,dc=org"

# Cleartext password: Trac
rootpw          {SSHA}yGq6aHM4w3Hf94hl4j+1rgO3HSGmmbVq
lastmod         on

# Path to the database files
directory       /var/local/db/tracldap
```

```
# 1.3.6.1.4.1.15527 is reserved. Do not hijack it
# Please see http://www.iana.org/cgi-bin/enterprise.pl

# Attribute type definitions
attributetype ( 1.3.6.1.4.1.15527.143
                NAME 'tracperm'
                DESC 'Trac Permission'
                EQUALITY caseIgnoreMatch
                SUBSTR caseIgnoreSubstringsMatch
                SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{32} )

# Class definitions
objectclass ( 1.3.6.1.4.1.15527.8
              NAME 'tracuser'
              DESC 'Regular user with Trac permission'
              SUP top
              STRUCTURAL
              MUST ( uid $ cn $ userpassword )
              MAY  ( tracperm $ sn $ description ) )
objectclass ( 1.3.6.1.4.1.15527.9
              NAME 'tracgroup'
              DESC 'Trac permission for groupofnames'
              SUP top
              AUXILIARY
              MAY  ( tracperm ) )

# ACLs (warning: give read access to anonymous LDAP connection)
access to dn.base="" by * read
access to dn.base="cn=subschema" by * read
access to filter=(|(objectclass=groupOfNames)(objectclass=tracuser)) dn.one="dc=example,dc=org"
       by group="cn=managers,dc=example,dc=org" write
       by * read
access to attrs=tracperm
       by group="cn=managers,dc=example,dc=org" write
       by self read
       by users read
       by anonymous read
access to attrs=entry dn.subtree="dc=example,dc=org"
       by * read

# Search indexing
index  objectClass,uid eq
index  cn,sn           eq,sub,pres,approx
index  member          eq
```

You should include this file from the main OpenLDAP configuration file, usually located here: `/etc/ldap/slapd.conf`. You need to include these definitions at the bottom of the file.

Configure your system logger

OpenLDAP errors are somewhat cryptic. You can find useful information in the log produced by the server.

It is very useful to compare requests made by standard utilities such as `ldapsearch` and the requests made by the extension:
If an ldapsearch request fails, blame your server configuration (or your directory content) not the Trac Ldap Extension

1. Add the following entry in `/etc/syslog.conf`

   ```
   # Log OpenLDAP
   local4.*                        -/var/log/openldap.all
   ```
2. Reload the syslog configuration

   ```
   /etc/init.d/sysklogd reload
   ```
3. You probably want to open a console and keep dumping the log messages:

   ```
   tail -f /var/log/openldap.all
   ```

Start up the LDAP server

1. Create the directory where the LDAP directory files will reside

   ```
   mkdir /var/local/db/tracldap
   ```
2. Start up the server

   ```
   /etc/init.d/slapd start
   ```

You should not get any error. If you get an error message (carefully check the log file), please fix up your LDAP configuration before resuming installation.

If everything is ok, shut down the server right now, because we need to initialize the LDAP directory

Initializing the directory

We need to create the top-most entry (the local root) of the LDAP hierarchical directory.

1. Copy the following LDIF data in a file, `init.ldif` for example:

   ```
   dn: dc=example,dc=org
   dc: example
   o: Trac
   description: Test directory for Trac
   objectClass: dcObject
   objectClass: organization
   ```
2. Then inject this LDIF data into the LDAP directory using the server tool. **Yes**, the server should be down at this very moment

   ```
   /usr/sbin/slapadd -b "dc=example,dc=org" -l init.ldif
   ```
3. At this point, you can restart the LDAP server

   ```
   /etc/init.d/slapd start
   ```

Now that the server is up and running, we can inject the initial directory entries that are expected by the extension unit tests.

1. Copy the following LDIF data in another file, `dirtest.ldif`

   ```
   # Group definition
   # Managers is a group that has permission to add and revoke Trac permissions
   dn: cn=managers,dc=example,dc=org
   ```

```
cn: managers
objectClass: groupOfNames
objectClass: tracgroup
member: uid=trac,dc=example,dc=org

# Group definition
# Users is a group of regular users
dn: cn=users,dc=example,dc=org
cn: users
objectClass: groupOfNames
objectClass: tracgroup
member: uid=joeuser,dc=example,dc=org

# User definition
# Trac is the 'software user' that manages the Trac permissions
dn: uid=trac,dc=example,dc=org
uid: trac
cn: Trac Manager
userPassword: Trac
objectClass: tracuser

# Special 'user': anonymous
# joker entry for non authenticated access
dn: uid=anonymous,dc=example,dc=org
uid: anonymous
cn: Trac Anonymous
sn: Anonymous
userPassword: no_use
objectClass: tracuser

# Special 'user': authenticated
# joker entry for any authenticated access
dn: uid=authenticated,dc=example,dc=org
uid: authenticated
cn: Trac Authenticated
sn: Authenticated
userPassword: no_use
objectClass: tracuser

# User definition
# Joe User is just a regular user
dn: uid=joeuser,dc=example,dc=org
uid: joeuser
cn: Joe User
sn: User
userPassword: anypasswd
objectClass: tracuser
```

2. Add those entries to the directory using the client tool. This won't work if the LDAP server is down

```
ldapadd -D "uid=root,dc=example,dc=org" -x -W -f direst.ldif
```

You'll be prompted for the user password, *i.e.* the password for user `uid=root`. This password is defined in the LDAP directory config file, here: "Trac"

At this point, you should be able to fully use the directory:

1. Search entries using an anonymous bind:

```
ldapsearch -b "dc=example,dc=org" -x objectclass=*
```

1. Search entries using an authenticated bind (password for trac is "Trac" too):

   ```
   ldapsearch -b "dc=example,dc=org" -D "uid=trac,dc=example,dc=org" -x -W objectclass=*
   ```

1. You can also add new entries and remove them if you like. But do not forget that the Ldap Extension unit tests expect the directory to be set up as described up to now

Clean up

If the test fails or some part of the installation procedure fails, you want to clean up the LDAP directory to restart from a clean environment.

1. Shut down the OpenLDAP server

   ```
   /etc/init.d/slapd stop
   ```
2. Remove the LDAP database files

   ```
   rm /var/local/db/tracldap/*
   ```
3. Reinitialize the directory (see above)

Troubleshooting

OpenLDAP server is very touchy, so double check your configuration files and your LDIF files if you get into troubles in the early setup stage.

Common errors

- `slapadd: could not parse entry (line=n)`
  This usually means that your initial LDIF file is malformed:
    - DOS vs. UNIX line ending mismatch
    - Extra trailing space

- `ldapsearch` returns no result
    1. Ensure that your base tree match the one defined in the LDIF file
    2. Try disabling the ACL (comment the rules and restart the `slapd` server)