

The OpenLDAP C++ API

Ralf Haferkamp
SUSE Research & Development
rhafer@suse.de

October 13, 2006



Novell®

Overview

- Started as a project for final thesis
- ISO C++ to be portable
- Wraps around async the OpenLDAP C library (libldap)
- Provides synchronus and asynchronus Interface
- Uses C++ Exceptions for Error Handling
- Used by YaST (e.g. User-Management)

Most Important Classes

- `LDAPConnection` and `LDAPAsyncConnection` for representing a Connection to the LDAP Server
- `LDAPMessageQueue` for receiving Results of asynchronous Operations
- `LDAPResult` and inheriting classes, representing the Results of LDAP Operations
- `LDAPException` for handling errors
- `LDAPEntry`, `LDAPAttribute` and `LDAPModification` to represent Entries, Attributes and Modifications
- `LDAPConstraints` representing connection properties

Synchronous Interface

- `LDAPConnection` implements a synchronous interface. (Block until results are received from the Server)
- Operations either succeed or throw an Exception (`LDAPException` or `LDAPReferralException`)
- `search ()` Method returns `LDAPSearchResults` Object to iterate over the Results.

Establishing and LDAP Connection

```
LDAPConnection lc("localhost");  
try {  
    lc.bind("cn=user,dc=example,dc=org",  
           "secret");  
} catch (LDAPException e) {  
    std::cerr << "Bind failed: " << e  
              << std::endl;  
}
```

LDAP Operations

- Easy to use to excessive use of default values
- A simple subtree search:

```
LDAPConnection lc;  
LDAPSearchResults *res;  
res = lc.search("dc=openldap,dc=org");  
LDAPEntry *e = res->getNext();  
std::cout << e->getDN() << std::endl;
```

- Errors trigger LDAPExceptions

Classes for Entries and Attributes

- LDAPEntry
 - Represents a single LDAP Entry
 - DN an List of Attributes
- LDAPAttributeList
 - Collection class to hold a list of LDAPAttributes
 - mostly a wrapper around std::list
- LDAPAttribute
 - Represents a single LDAP Attribute, its type and its values
 - The Attribute's values are stored in a StringList

Example

```
LDAPEntry e("ou=example,dc=example,dc=org");

LDAPAttributeList *al = new LDAPAttributeList();

LDAPAttribute a("objectClass");
a.addValue("organizationalUnit");
al->addAttribute(a);

LDAPAttribute a("ou");
a.addValue("Example");
al->addAttribute(a);

e.setAttributes(al);
```


Entry Modifications

- Modification to single LDAP Attributes are described by `LDAPModification` objects:
 - `LDAPAttribute` and the type of modification to apply
`OP_ADD`, `OP_DELETE`, `OP_REPLACE`
 - Very similar to the `LDAPMod` struct of the C-API
- `LDAPModList` is used to group several `LDAPModification` objects

Modify Example

```
LDAPAttribute attr("description");
LDAPModification(attr, LDAPModification::OP_DELETE);

LDAPModList *mod=new LDAPModList();
mod->addModification(LDAPModification(newattr,op));

try {
    lc.modify("cn=user,dc=example,dc=org", mod);
} catch (LDAPException e) {
    std::cerr << "Bind failed: " << e << std::endl;
}
```

Asynchronous LDAP Operations

- `LDAPAsyncConnection`
- Results handled via the `LDAPMessageQueue` class
- used as the base for `LDAPConnection`

Example

```
try {
    LDAPMessageQueue *mq = lc.add(&e);
    LDAPMsg *res = mq->getNext();
    if ( res->getMessageType() == LDAPMsg::ADD_RESPONSE )
    {
        int resCode = ((LDAPResult*)res)->getResultCode();
        if ( resCode != LDAPResult::SUCCESS ) {
            cout << "Add failed: "
                 << ((LDAPResult*)res)->resToString() << endl;
            cout << "Additional Info: "
                 << ((LDAPResult*)res)->getErrMsg() << endl;
            exit(-1);
        }
    }
    delete(mq);
} catch (LDAPException e) {
    std::cerr << e << std::endl;
}
```

Misc. other Features

- Class LDAPUrl for working with LDAP Url
- Methods and Classes for querying and parsing the Schema of a Server
- Automatic Referral Chasing (the callback mechanism used to reauthenticate with a target server needs work)

Missing Features / ToDo

- SASL authentication
- LDAPS / LDAPAPI support
- LDIF support
- Improve memory handling (User should need to call delete on Objects created by the library)
- Probably remove the need for libldap and call into liblber directly.
- Patches welcome

Thank You. Time for Questions