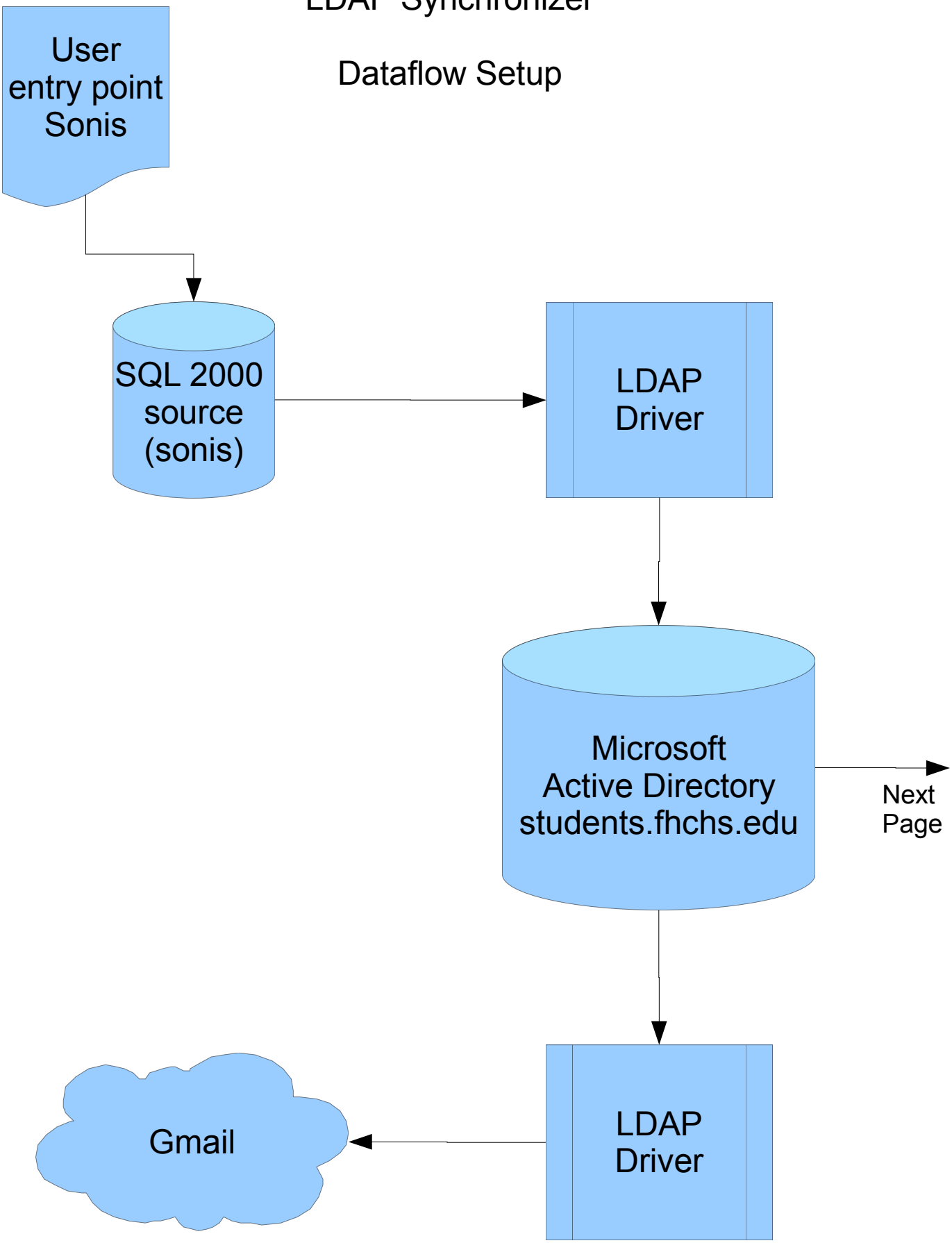
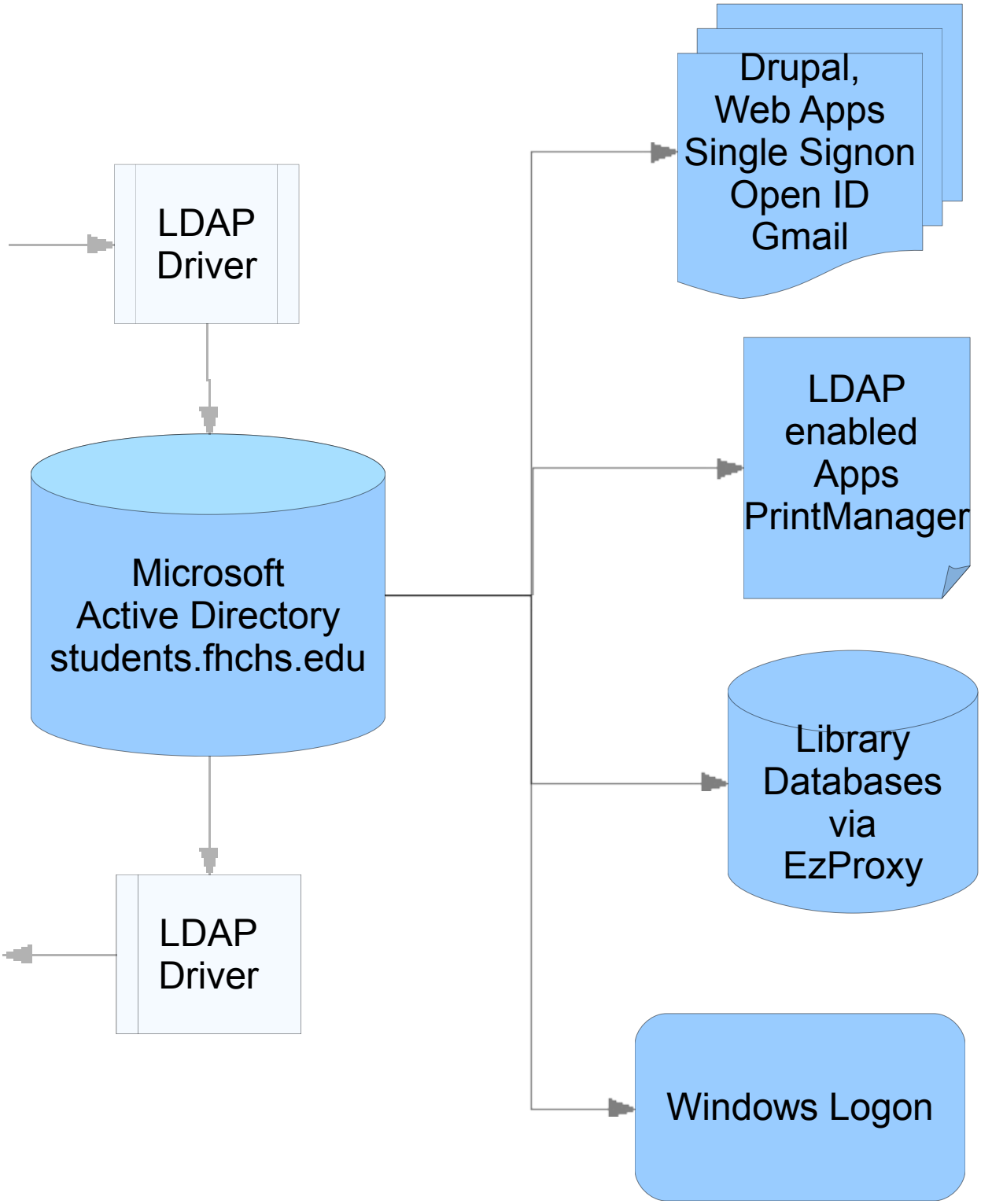
LDAP Synchronizer

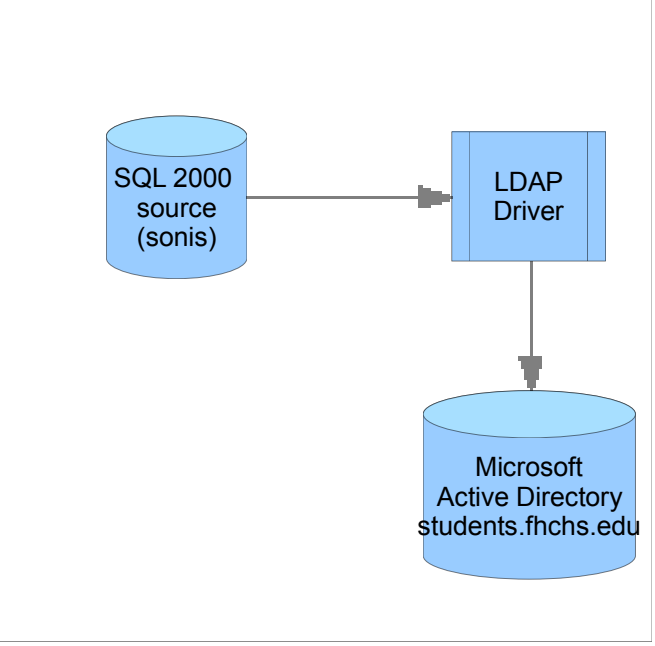
Dataflow Setup



LDAP Synchronizer

User Application connection





LDAP Synchronizer: Step 1

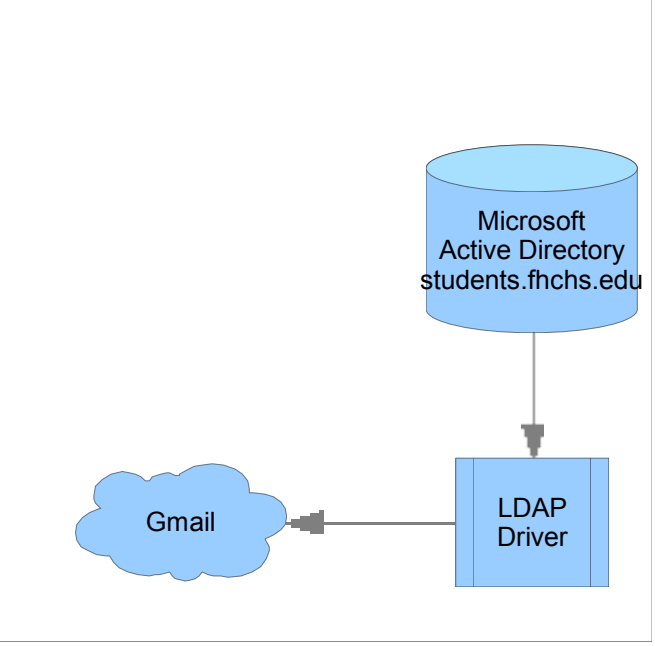
- Conversion of any SQL server 2000 data into Microsoft Active Directory Users, or Security Groups.
- Accounts mapped from sonis to AD by field assignment.
- Synchronizes data based on user unique ID from the database source to AD
- Account data is not actively synchronized. Synchronization will be scheduled at least once a day. Testing will give an idea how often the synchronizer can run. Synchronization one direction from sonis to AD.

How it runs

- Task scheduler calls the LDAP driver, giving the parameters for a saved mapping and the type of operation.
- LDAP driver executed the selected operation and creates a log.
- The log file generates a history of successful transactions, warnings and errors. It is currently undecided how to save the logs.

Setup for students.fhchs.edu

- soniswebdatabase.name table and soniswebdatabase.address table are used to fill out the information for AD users.
- soniswebdatabase.nmcrs table and a collaboration of other tables are used to create Security groups for LDAP enabled application permissions. LDAP uses on the horizon; Print manager +, gmail via single signon, local logins...



LDAP Synchronizer: Step 2

- Conversion from AD to Gmail
- Student users created as xx555555555@students.fhchs.edu
- Student email accounts aliased as first.last@students.fhchs.edu, if there are duplicates, combinations of the middle and first name are abbreviated to get a unique name.

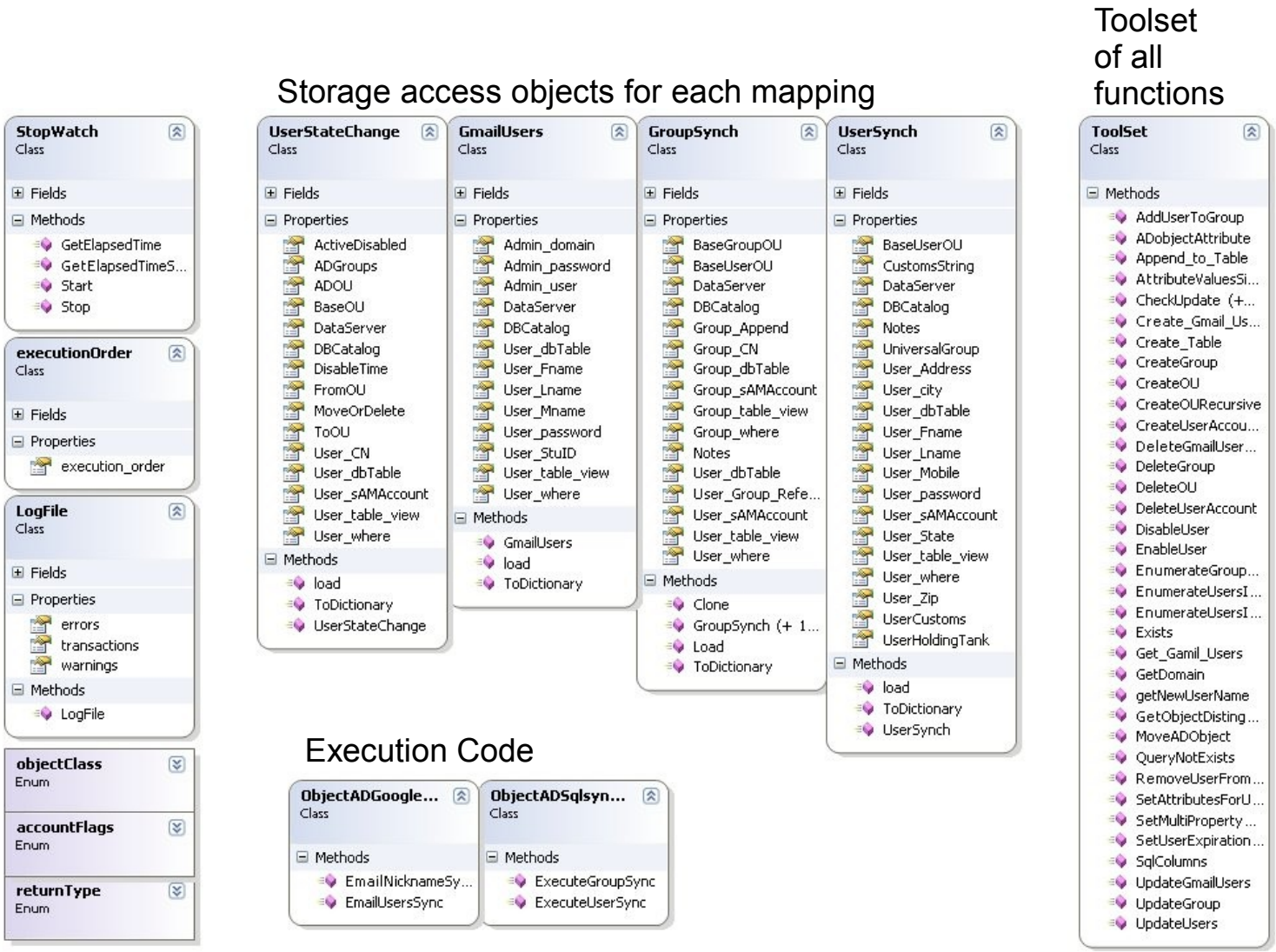
How it works

- Same as LDAP driver Step 1. Input file and operation from the task scheduler start a background process.

Setup for students.fhchs.edu

- User information is pulled from the user account in active directory.
- Users are created only for users in a curent program. (only new students will get an email account).
- Account detention details are undecided.
- Currently accounts will be updated with new names if the user record changes ie.. marriages. An additional account alias will be made for the new name but their original will not be deleted (first.last@students.fhchs.edu).

utils/Toolset.cs
Basic code outline



A quick legend. The hands holding a paper are stored values. The small purple boxes are actions that can be done with the data. Large boxes represent classes.

The execution code boxes are the massive methods which do the synchronization between AD, SQL, and gmail.

The toolset contains all the necessary utilities for working with SQL, Gmail, AD.

The top four classes store the properties for the program while it is running.

Password Resets

• Web based page (.asp NET) which resets LDAP password, sonis password, gmail password ? Angel integration?
• Multiple pieces of information from user profile in sonis required.
• New passwords will need to match AD strong password requirements.

How Teachers Can Send an Email

• Emails will be sent through outlook. Each computer will need the students.fhchs.edu LDAP address book attached.
• Emails can be sent to the group name associated with any class and other designations created by the tech department.
• A survey was taken to determine the most useful email groups to create. Some requested groups are divisions by department, class, masters, BS, AS

Student User Flow

• Gmail login performed by single sign-on at the student portal.
• Each application will have a different way of tying into Active Directory and our student portal.

LDAP Driver Requirements

• LDAP driver requires the account running it to have permissions on the AD server to create, delete, update; groups users and organizational units.
• The account running LDAP driver must have permissions on the DB server for the database used. As well to create temp databases.
• LDAP driver is currently only sql server 2000 compliant
• LDAP driver will fail if field names which it uses in the database are changed. The save files will need to be updated.

Code Design

• LDAP driver is split in 4 main files.
    - Program.cs holds the command line logic.
    - Form1.cs holds the gui code
    - utils/toolset.cs holds the entire logic
    - utils/arguments.cs holds the command line parser

Credits

• The LDAP driver is written in c# for Florida Hospital College Of Health Sciences.
• Code by Michael Neubrander, Travis Wooley
    - arguments.cs by R. Lopes no attached license
    - A free non-restricted for commercial use version is available
• Graphics and UI design Paul Martin