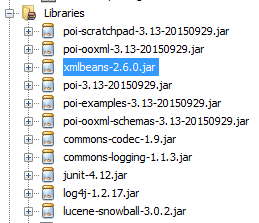
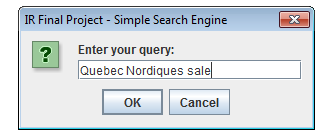
README:

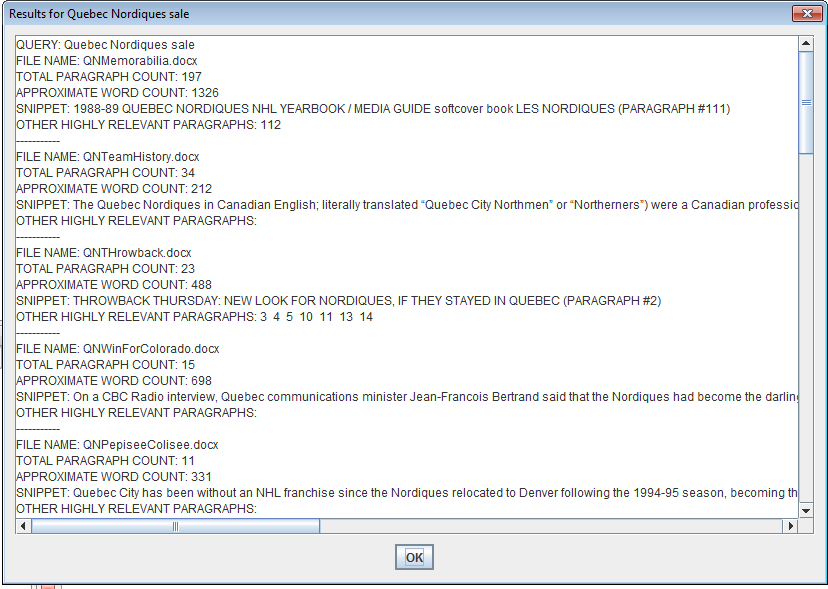
1. Extract all files and directories from the zip file into a new folder (this was already done on the flash drive submission but not the e-mail submission).
2. Create a new blank java project and import the two JAVA files found in the SRC folder to the default package generated by the project (InvertedIndex.JAVA and SearchEngine.JAVA).
3. Navigate to the “Stemmer” folder in the extracted directory and add the lucene-snowball-3.0.2 JAR file into the new project’s library. This JAR file is required in order for Porter’s Stemming Algorithm to work in the program.
4. Navigate to the “poi-3.13” folder. This contains all of the files of the Apache POI 3.13 library, which is required to read the docx files that comprise the corpus. Move all available JAR files in the “poi-3.13” directory, the lone JAR file in the “ooxml-lib” directory, and all JAR files in the “lib” directory into the project library. There should be ten JAR files from the Apache library added to the project. The project library should now look as follows:



1. Save the project. The program should now be able to be ran from the command line by compiling both source files and running SearchEngine.JAVA. The main function expects one command line input argument, which is the absolute path to the directory labeled “Corpus” from the extracted files (i.e. project was tested using “H:\Information Retrieval and Org\Project\Corpus”). It is important that the directory structured from the provided zip remains the same and that the path of this directory is specifically passed as the first command line argument (args[0]) in order for the program to function correctly.
2. Upon running, the Inverted Index will be constructed from the contents of the Project folder. A few seconds after, a new window will appear prompting the user for input. Type in a query and click “OK”.
3. Upon running, the Inverted Index will be constructed from the contents of the Project folder. A few seconds after, a new window will appear prompting the user for input. Type in a query and click “OK”.



1. Upon clicking “OK”, a new window displaying the results to the query will appear. The precision and recall for a particular query can be found by scrolling down to the bottom of the window.



1. Clicking “OK” or exiting out of the window will open the query window seen in step 6, where another query can be entered and its results can be viewed.
2. When you are finished entering all necessary queries, click “Cancel” or simply exit out of the query window on the screen. The result window will pop up one last time, showing a list of all of the results from the queries that were entered during the session. The results for the entire session is also saved in a text file “results.txt”, which is written to the same directory containing the Java project.