**CIS 345: Lucene XML Search Application**

Fall 2014 - Weimin He

*Joshua Evenson, Jonathan Meer*

**Course Project**

Due Date: 11:55pm, Monday November 24th, 2014

1. **Introduction**

This application was built to provide the functionality to search a large number of XML files. It utilizes a 3rd party library called Lucene from Apache that we use to index our XML objects. This indexing greatly increases our search times.

The application is made up of two parts: a Windows Form app for the user interface and the query building, and a Java project that handles the indexing of documents and searching the index with the provided query. In order to utilize the Java program from the Windows Form app, we utilized IKVM.NET to compile the Java JAR file into a DLL which is referenced in the Windows Form app.

1. **Java Project Documentation**

Our Java Project is made up of five files, three of which are used in the indexing and searching of XML files.

Driver.java

This file is only used when testing the application in the Java development environment. It houses the main method that calls the GO method in LucemeXMLProcessor.java

HelloLucene.java

This file was obtained from <http://www.lucenetutorial.com> in order for us to learn how Lucene operated. We left it in the project as a reference on how to use Lucene.

LuceneXMLProcessor.java

This file is where the bulk of the work happens. It contains the GO method, which accepts a directory and a query from the Windows Form apps. With that directory, it will build the index of XML documents from all the files in that directory. Once that index has been created, it is added to a Map (key value pair collection) that stores the index for future use. Before a new index is created, it checks to see if it already exists in the Map. That way if the index already exists, a new one does not need to be rebuilt, saving a lot of time.

Once the index has been built, the query is run. We take the results of the query and turn them into key value pairs and insert them into a Vector. That Vector is returned from the method and interpreted by the Windows Form app for presentation to the user.

LuceneDocumentBuilder.java

In order for Lucene to build an index, it must take data from a source and turn it into a document. That document is then added to the index for Lucene to search. The first step is to retrieve a NodeList from the given XML file using the XMLParse.java file. Iterating the NodeList we create a field for each child element. The key is the node name and the value is the node value. By using the node name as the document field title, we do not have to define a document specification for each type of XML file. The same code can be used on both of our XML schema objects, presented in part 4.

XMLParser.java

This file was created to resolve an import redundancy between the Lucene Document class and the Java Document class. Since Lucene has a class called Document to build indexes, and we needed to use the Java Document class to read XML files, the algorithm that reads XML files was separated into a different file. The parseXMLFile function is passed a file path and retrieves a NodeList to pass to the LuceneDocumentBuilder.java file.

1. **Windows Forms Project Documentation**

The User Interface Consists of four forms.

AppStart

The start screen for the app. The user will choose which type of object to search for.

CustomerSearchForm

The form users use to search for customers. Each field for a customer is displayed and the Lucene search query that will be passed to the Java project is built in the text box above. Clicking search will execute the indexing and search in the Java project and the results will be shown on a data grid view on the SearchResults form.

BookSearchForm

The form users use to search for books. Each field for a book is displayed and the Lucene search query that will be passed to the Java project is built in the text box above. Clicking search will execute the indexing and search in the Java project and the results will be shown on a data grid view on the SearchResults form.

SearchResults

The results of the search will be displayed in a data grid view.

1. **Objects**

Each object type has 50 XML files in their directory, each file containing 100 objects.

|  |  |
| --- | --- |
| Book |  |
| Customer |  |

1. **References**

* Lucene - <http://lucene.apache.org/>
* IKVM.NET - <http://www.ikvm.net/>
* Lucene Tutorials - <http://www.lucenetutorial.com/>
* Git Hub Repo (Java Only)- <https://github.com/jevenson/CIS345Lucene/>