Project #2 – Composite Text Analyzer, version 2.6 due Monday, Oct 7th

Purpose:

This project develops a program for making queries into a set of ASCII text files with associated metadata. The metadata consists of one XML file for each text file. The XML describes properties of the associated text file, specifically: name, time-date of last modification, version number, description, dependencies, size, and zero or more keywords. Queries are made by extracting information from the XML and also by searching the text file for specified strings.

As part of this project you will also build a tool for creating a file's metadata. The tool constructs an XML file and inserts elements to describe the file's attributes as cited in the preceding paragraph. The tool accepts description text and lists of dependencies and keywords. There may be metadata files not associated with any text file. Ask about this in class.

The purpose of this text analysis is to support management of large sets of document files. Projects #3 and #4 will use the results of this project to help construct a Document Vault and Project #5 will explore using this facility as part of a Software Repository.

Requirements:

The Text Analyzer:

1. **(5-CaL) Shall** be implemented in C# using the facilities of the .Net Framework Class Library (FCL), version 4.5, as provided in the EECS clusters. Use of a Graphical User Interface is optional[1], and may be implemented in either WinForms or Windows Presentation Foundation (WPF).
2. **(3-R) Shall** accept a specification of files using file patterns and a path from the command line and queries, as described in the next two requirements.
3. **(6-R) Shall** accept one or more command line text queries of the form /T"some text" /T"some additional text" and return the fully qualified name of every file containing that text. The Analyzer **shall** accept command line switchs /A and /O to indicate that the search must require matches of all text strings or require matches of at least one text string. You are only required to search text files[2], e.g., .txt, .cs, .dat, etc.
4. **(6-R) Shall** accept a command line metadata query of the form /Mtag1,tag2,… where tag1, tag2, … are tag names of metadata elements, and return the contents of each element of the metadata for every file in the file set. If a file does not have an associated metadata file, the analyzer **shall** emit an error message and continue processing the remaining files.
5. **(2-R) Shall** accept a command line recursion command of the form /R. If present on the command line /R indicates that the file set shall include all files in the directory tree rooted at the specified path. If not present the file set consists of only the files in the specified directory.
6. **(3-R) Shall** provide a separate metadata tool that creates a metadata file for a specified file. The tool input consists of a fully qualified file name, description text, and lists of dependencies and keywords. These inputs are provided by the user on the tool's command line using switches /T, /D, and /K.
7. **(5-CaL) Shall** submit a compile.bat file that builds your project using a grading machine and a run.bat that demonstrates you meet all the functional requirements: #2-#6.

Your compile.bat and run.bat files compiles and runs your project without the user needing to specify any information, e.g., you are providing default values for path, patterns, and queries. Please make the path be the relative path to your project and the files that make up your project.

[1] Not recommended.

[2] You will receive a 5 point bonus if you fully satisfy requirement #3 for .doc files as well as text files. To do that you need to make a reference to Microsoft.Office.Interop.Word.