

Project #1 – Text Search Component

due Tuesday, March 17

Purpose:

For this project we will develop recursive text finder components that can run on Windows and Linux. The purpose of the finder is to identify all files in some directory tree that contain one or more specified strings¹. In this project you will develop a program that searches a directory subtree and reports any files that contain a string passed to the command line.

The Windows component will be developed using Microsoft's Component Object Model (COM) technology. For Linux, which does not support COM we will build a C++ library that exposes its functionality through an interface and object factory and uses reference counted life-time management. We will see that COM does that too, and more. These components will be used through scripts, a C++ client on both platforms, and also a C# client on Windows.

As part of this project you will expand a FileSystem² facility that supports file and directory analysis, and navigation of directory trees, by adding finding system time, and reporting system errors. You will then design and implement applications that search a path specified on the command line and report any files that contain a string, also specified on the command line.

Because the working set is potentially large, and we need to make persistent some form of the analysis, there are several interesting design questions for you to consider. Feel free to ask about these in class.

Requirements:

Your TextFinder project:

1. **shall** use standard C++ and the standard library, compile and link from the command line, using `g++` within the NetBeans or Eclipse IDE and Visual C++ in the Visual Studio 2013 IDE.
2. **shall** add code to the C++ FileSystem package for the Windows and Linux environments, as needed to support your project. The external interfaces of these packages **shall** be identical on both platforms.
3. **shall** provide a console application that analyzes all of the files and directories found on a path specified on the program's command line, and reports any files that contain one or more strings specified on the command line. Please implement a `/O` option indicating that a match is found if any one of the strings are discovered. Absence of the `/O` option indicates that all strings must be found.
4. **shall** build the searching functionality on Windows using the COM technology, via ATL facilities provided by Visual Studio 2013. This results in a dynamic link library configured with the COM protocols.
5. **shall** build the searching functionality on Linux using native C++, providing an interface, implementation, and object factory, built as a dynamic link library. This component **shall** implement reference counting and functions `AddRef()` and `Release()` to support that.
6. **shall** provide native C++ library clients that provide user access to the search facilities for both platforms, and shall also provide a C# client for Windows.
7. **shall** provide a PowerShell script on Windows and a Bash script on Linux for testing and to demonstrate that you met each of the requirements specified here.

You will find it helpful to look at the Man pages for System Calls on your Linux system. Those describe the semantics of each call and the header files you will need to include.

A ten point bonus will be awarded for successful implementation of the search functionality so that each file is analyzed on its own thread. You may use C++11 threads and locks, or, on the Windows platform you may use I/O completion ports. The bonus will depend on you executing performance demonstrations for single and multiple thread operation.

¹ It would be nice to accept regular expressions, but that is not required for this project.

² <http://ecs.syr.edu/faculty/fawcett/handouts/CoreTechnologies/Cpp/Code/FileSystem-Windows/>,
<http://ecs.syr.edu/faculty/fawcett/handouts/CoreTechnologies/Cpp/Code/FileSystem-Linux/>