**Project #4 – Remote Test Harness** due Tuesday, May 01
version 2.3

Purpose:

This project provides a facility for remotely executing a series of tests, defined by an XML configuration file. The configuration file defines a sequence of test libraries that are created using the results of your first project. The test harness provides an input queue to receive test configuration messages. For each test configuration it runs all the tests specified in a separate process. The test harness spawns a process for each test configuration and receives the results back from that process to pass back to the user.

The activities in the Test Harness are as implemented in Project #3. To that we add message-passing communication that supports executing tests on a remote machine from code that initially resides on the user’s local machine. Also, this project provides a rich user interface for configuring tests and exploring the results of those tests.

Requirements:

Your Remote Test Harness:

1. **shall** use standard C++[[1]](#footnote-1) and the standard library, compile and link from the command line, using Visual Studio 2010, as provided in the ECS clusters and operate in the environment provided there[[2]](#footnote-2).
2. **shall** use services of WinForms using C++/CLI or Windows Presentation Foundation, written in C#, communicating through a C++/CLI bridge to all of the client functionality defined in one or more dynamic link libraries.
3. **shall** provide the Test Harness facility you developed in Project #3, along with a WinForm or WPF client and socket-based message-passing communication facility.
4. The Test Harness Client **shall** provide means to browse for and select configuration files for testing[[3]](#footnote-3). It **shall** send that to the Test Harness input queue, accessed through the communication facility you provide. The Test Harness **shall** request files from the Client that match its test configuration, and after receipt, **shall** process them as described in the Project #3 specification[[4]](#footnote-4).
5. The Test Harness **shall** send test results back to the client for display. These results **shall** consist of a formal result that includes Test Name and ID, test author, date of testing, and success status of each package test. The results may also contain an informal part that consists of whatever logged output the test designer implemented. The client **shall** provide means to view both the formal and informal parts in two different views.
6. Please submit all the code for your project in a zip file that also contains a Visual Studio solution and a compile.bat and run.bat that will compile and run your project, demonstrating that it meets all requirements specified here. Please provide tests of your design for the Tokenizer and SemiExpression packages provided here: <http://www.lcs.syr.edu/faculty/fawcett/handouts/CSE687/code/Parser/>
or create other simple demonstration tests.

You will need to be familiar with the following:

* Socket Communicator: <http://www.lcs.syr.edu/faculty/fawcett/handouts/CoreTechnologies/SocketsAndRemoting/code/SocketCommunicator/>
* WinForms:
<http://www.lcs.syr.edu/faculty/fawcett/handouts/CoreTechnologies/WindowsProgramming/code/FormWithCppThread/>
* WPF:
<http://www.lcs.syr.edu/faculty/fawcett/handouts/CoreTechnologies/WindowsProgramming/code/>
1. This means, for example that you may not use the .Net managed extensions to C++ or C# except for the GUI. [↑](#footnote-ref-1)
2. VC++ 2010 is available in all the ECS clusters. [↑](#footnote-ref-2)
3. Three config files will constitute an adequate demonstration. [↑](#footnote-ref-3)
4. You may also satisfy this requirement by providing a utility to copy files from the client to the Test Harness before testing so that the Test Harness does not have to request files from the client. An earlier version asked you to browse for dlls and construct a config file to send. That also satisfies this requirement. [↑](#footnote-ref-4)