

Project #4 – Dependency Analyzer

due Monday, Nov 17th

Version 1.5 (changed due date)

Purpose:

This project develops a Dependency Analyzer that uses the analysis engine you developed in Project #2. It is charged with finding all types and/or packages that are dependent on a specified type or package. It must do this for source code file sets that may reside on one or more directories in local or remote machines. This will require you to develop both server and client programs where the server may reside on a remote machine.

Understanding these dependency relationships is essential for software development, testing, and maintenance. For example, we may wish to determine the impact of making modifications to a specified package. Which other packages may be affected? How can we develop a test plan that minimizes retest when we find defects requiring changes to this package's source code?

Requirements:

The Dependency analyzer:

1. **Shall** compile¹ and run with Visual Studio 2013 and the environment provided in the ECS clusters.
2. **(1) Shall** accept one or more specifications of files to analyze on the command line using addresses and file patterns.
3. **(1) Shall** provide an option², which when activated, causes search for the file specification to be analyzed recursively on the directory sub-tree rooted at each specified path.
4. **(5) Shall** find dependencies between all types defined in the complete file set (from all machines).
5. **(2) Shall** find dependencies between all packages in the complete file set (from all machines).
6. **(2) Shall** create an XML file that captures the type and package dependencies found above.
7. **(2) Shall** display, in a Graphical User Interface, the dependency relationships found above. **Shall** provide an option to display relationships by making LINQ queries into the XML created for Requirement #6 from a previous request.
8. **(1) Shall** provide an option to display only type dependency information and provide an option to display only package dependency information.
9. **(2) Shall** provide a server process that carries out the type and dependencies analyses and a client process that requests specific analyses and displays the results. Note that these dependencies may span more than one server³.
10. **(2) Shall** use Windows Communication Foundation (WCF) for all communication between client and server processes or machines. Communication **shall** be based on message-passing.
11. **(2) Shall** use Windows Presentation Foundation (WPF) for all User display. **Shall** use child components to implement most of the client processing. The GUI should focus on Presentation, not processing.

¹ Please use C# for all development in this course, unless specifically directed to do otherwise.

² Options can be implemented with WPF radio buttons or check boxes.

³ That means that all servers share the same type data. We'll discuss this in class.

12. **Shall** provide compile.bat and run.bat files that build your application and that demonstrate that you met requirements #2 - #11 above.
13. **Shall** supply all source code, project, and solution files for Visual Studio 2013.