



Analyzing C/C++ code from Visual Studio in the Clang module

May 2016

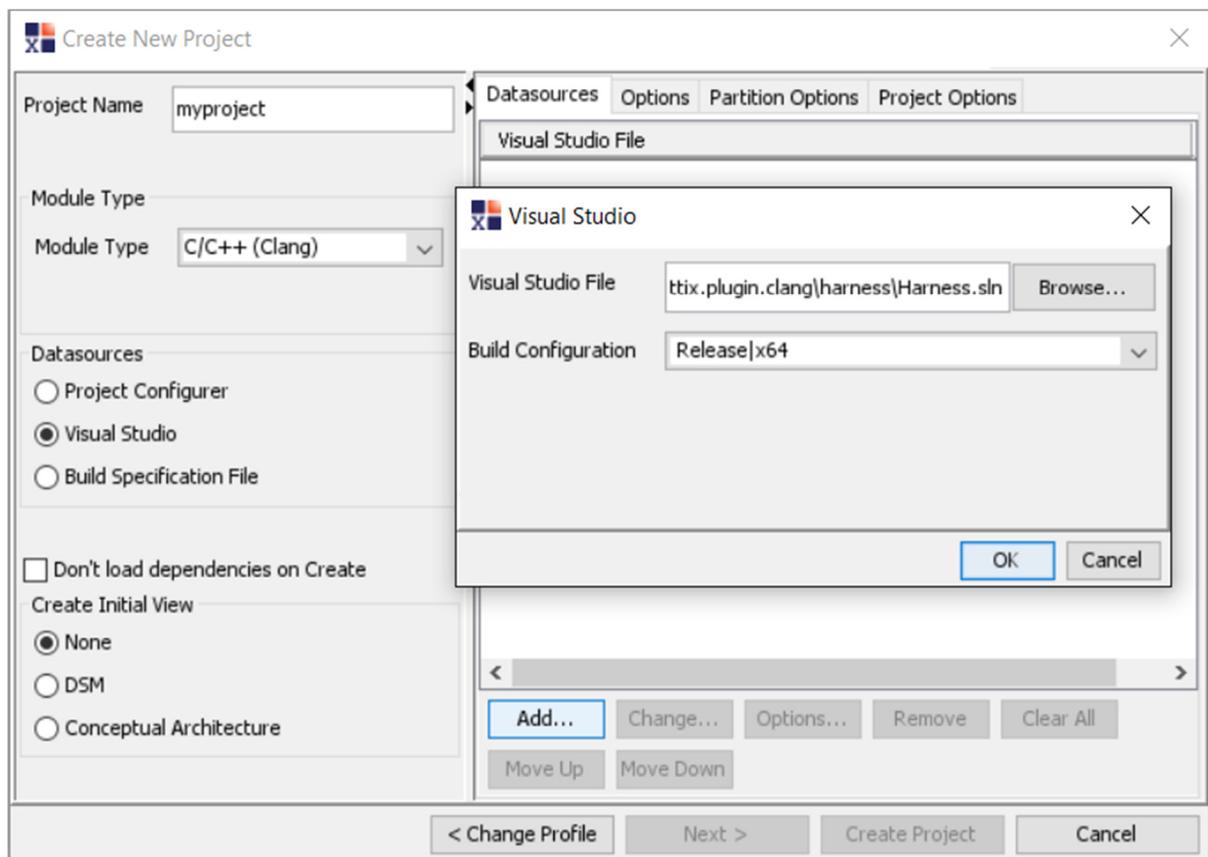
Introduction

The Clang module can be used to analyze C/C++ code in Visual Studio solution and project files. Since the studio project files contain the options to compile and link the executable, the Clang module can use that information to analyze and extract the dependencies.

Create your Project from Studio Files

Note that Lattix expects a Visual Studio environment when loading in the solution or project files. Therefore, you must have Visual Studio installed on the machine where Lattix is being run.

1. Click on **New Project** on the home page to bring up the **Create New Project** dialog.
2. Select **C/C++(Clang)** for **Module Type**.
3. Choose the option **Visual Studio** for **Datasources**.
4. Click on **Add** and specify the solution file or the project file(s), along with the Target.



5. Click on **Create Project**.

Check for Errors

Check for errors by bringing up the Clang Reports using **Reports->Clang Reports...->Diagnostics by File**. Please pay particular attention to **fatalerrors**, which are typically caused by missing include files. Most other errors will normally not have any significant effect on the analysis. Fix the errors and re-create (or update) the project.

Hint: To minimize the time, remember that you can test by compiling just a single source file that is giving you errors. The **Options** tab on the **Create New Project** dialog has an option called **Include files matching these names**. Simply enter the full path name of the file and then only that file will be compiled.

Analyze

You are now ready for further analysis using Lattix Architect.

1. Run the checkers to find architectural issues.
2. Run Clang reports for a variety of include file analyses including unnecessary includes.
3. Examine numerous metrics including stability, cyclicality and coupling.
4. Use a DSM or CAD view to identify problematic dependencies, apply partitioning to discover and reason about architecture, and re-organize structure to reflect components and layers.
5. Publish to Lattix Web to chart trends, diff builds, and display violations.

Additional Resources

A well understood modular architecture may be the most effective technique for improving the quality and maintainability of complex code bases. At Lattix we have helped some of the leading companies in the world improve the quality of their software and we can help you achieve the same results.

The Lattix Customer Portal contains additional white papers, customer case studies, and articles (<https://cp.lattix.com/whitepapers>) as well as demos and webinars (<https://cp.lattix.com/livedemos>). You can also check out our how to videos (<http://lattix.com/videos>).

If you have questions or comments, you can contact us at support@lattix.com. To try Lattix Architect on your code, email sales@lattix.com or call 978-664-5050.