



Analyzing C/C++ code on Linux using LXBuild and the Clang module

May 2016

Introduction

The Clang module analyzes C/C++ code for a particular **build specification**. A build specification specifies each source file and how it is compiled. A build specification also specifies which of those files are linked together. **LXBuild** is a Lattix Architect utility that monitors the build process and generates such a build specification.

Generate a Build Specification File

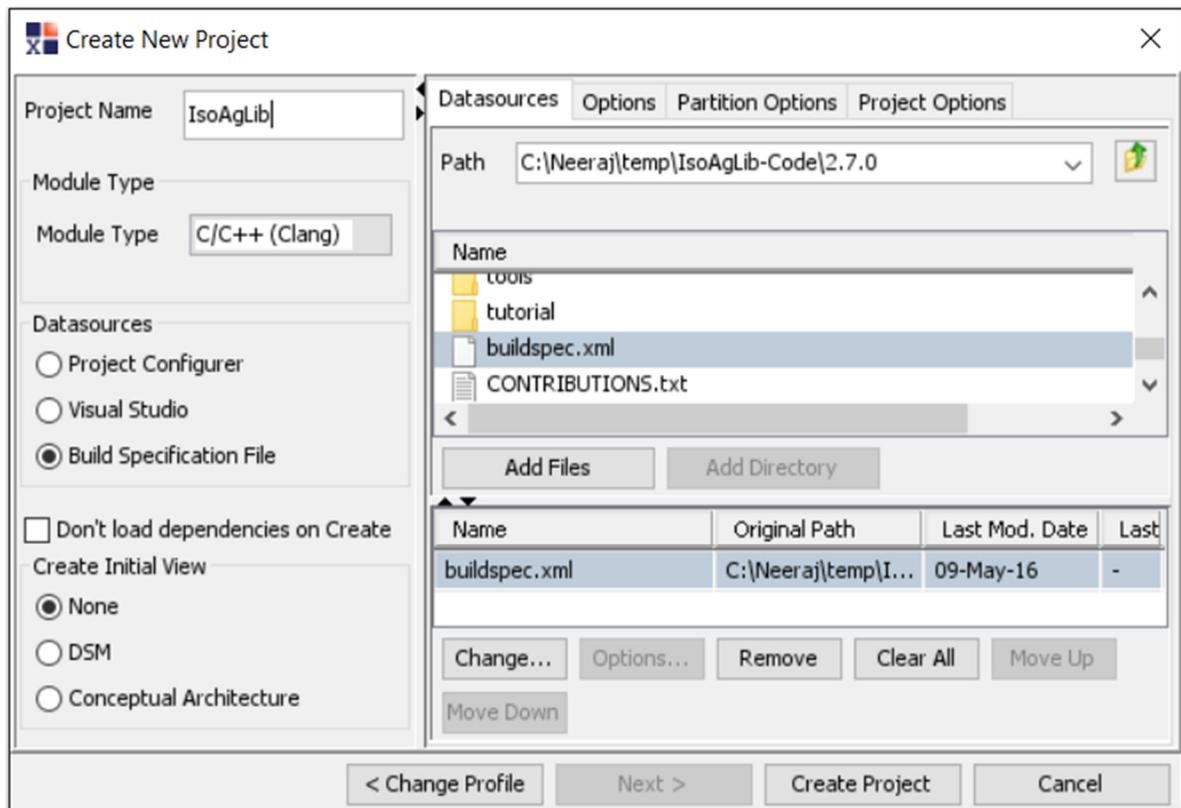
Run **LXBuild** and specify the command to build as an argument. For example:

```
➤ lxbuild make
```

You will see a new file called `buildspec.xml` when the build completes. This file contains the specification of the build. It consists of units, corresponding to each link unit. Each unit contains the files and the options for compiling them.

You can read more about this utility at <https://cp.lattix.com/help/v10/toolsmanual/lxbuild>.

Create your Project



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 **Build Specification File** for **Datasources**.
4. Navigate to the build specification file that was generated using **LXBuild**.
5. Click on **Create Project**.

Check for Errors

Check for errors by bringing up the Clang Reports using **Reports->Clang Reports...->Diagnostics by File**. The errors that cause analysis inaccuracies are typically from missing include files. Fix the errors and re-create (or update) the project. Note that you can specify additional include directories in the **Options** tab in the **Create New Project** dialog.

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.