

Extending Eclipse with Compiled Languages Other than C/C++

Position Paper for Eclipse Language Workshop

Doug Gaff

10/12/05

I'm interested in Eclipse language support for compiled languages like C, C++, Fortran, and Ada. I'm motivated by the needs of embedded software developers to write device software in multiple languages, typically because their applications often include vendor-supplied middleware, proprietary or open-source operating systems, and/or legacy code from previous projects, all of which can be in written different languages. I would like to see Eclipse become known as a general development environment for multiple languages to the same extent that it is known as a Java development environment. The closest parallel is probably Visual Studio, with the obvious caveat that Visual Studio supports languages and operating systems only of interest to Microsoft, whereas the goal of Eclipse should be to enable any compiled language of interest to the wider community.

I would like to see the following capabilities. This list is representative and not exhaustive.

- A common infrastructure for debugging without bias towards a specific language.
- A generic debug engine interface that doesn't assume GDB as the debug engine.
- The ability to seamlessly cross language boundaries during debugging.
- The ability to define and build complex mixed language projects.
- The ability to edit source files from multiple languages using a common look and feel editor.
- Comparable browsing, refactoring, and code analysis capabilities built upon a common framework for each language.

The overarching goal for language support should be a common, language-independent infrastructure for the entire edit-compile-debug lifecycle and its associated subtasks. Support for specific languages should be relatively simple implementations of the framework with minimal code duplication. The design goal should be to maximize changes made to the platform so that specific language editions don't have to be major projects unto themselves. Each language should look for areas where commonality exists as their projects evolve to ensure that common code moves into the Platform. Finally, the infrastructure should foster collaboration between language projects in order to encourage the development of solutions for more complex multi-language use cases.