

CodeCompass

An Open Software Comprehension Framework

Zoltán Porkoláb, Tibor Brunner, Márton Csordás, Máté Cserép, Anett Fekete, Endre Fülöp, Gábor Horváth
Ericsson Ltd. Eötvös Loránd University, Budapest

<https://github.com/Ericsson/CodeCompass> (live demo is available)

1. Introduction

CodeCompass is an open source framework that supports code comprehension. The information is gathered from all available resources: the source code, build commands, Git repositories, software metrics, etc. It provides wide scale textual and visual information displayed on a common WEB based GUI or queried by a Thrift API. Its functionality is extensible by plugins which may provide: a parser, a service layer, a database model and WEB GUI modules.

3. Incremental parsing

Parsing is the most time consuming action. However, code modifications are detected and only the affected translation units should to be reparsed. Individual users may have small code changes. These code editions can be modeled as database layers over a baseline parsing.

4. Typical comprehension workflow

1. Fast feature location

- Textual search
- Log-base search

2. Extend knowledge

- Info tree
- CodeBites view
- Function call chains
- Browsing history

3. Validate knowledge

- All references
- Architectural info
- Version control info

5. Features

- Macro expansions
- Symbol search
- Log output fuzzy search
- Entity info collector
- Function call diagram
- Class diagram
- C/C++ module diagram: header providers, object users, directory relations
- Git history browser
- Pointer analysis
- Displaying compiler generated functions
- Call chain via function pointers
- Metrics
- CodeBites

6. Future plans

CodeCompass currently provides a snapshot view i.e. one parsed version of the project is stored in a read-only database that can be accessed via a Thrift API. Language Server Protocol (LSP) is a Microsoft initiative for connecting an editor with a language smartness provider, like CodeCompass, by a similar interface. We are going to implement the LSP protocol for CodeCompass.

2. Database model

Parsers collect information about the project and store them into a relational database in an extremely compressed form. Storing all AST nodes in the database would be very inefficient (~1:1000 ratio between source code and DB size). By storing only the ones with names, as these are interesting, we achieve a ~1:30 size ratio considering C++ related tables.

Parsing the LLVM code base (320MB) itself results in a 21GB database.

Language related persistent information is available via an ORM tool. Its simplified schema is shown to the right.

