



Dump Lexical Contents of Class Definition

Course: Compiler Design

Course code: CS363IA

Computer Science and Engineering

Introduction

This project involves extending the Clang compiler by introducing a new frontend flag, `-fdump-class-extents`, which enables the emission of lexical extents for all class-like definitions (such as classes, structs, unions, and template classes) encountered during compilation.

When this option is enabled, Clang traverses the Abstract Syntax Tree (AST) and prints the fully qualified name of each class, the absolute file path where it is defined, and its starting and ending line numbers.

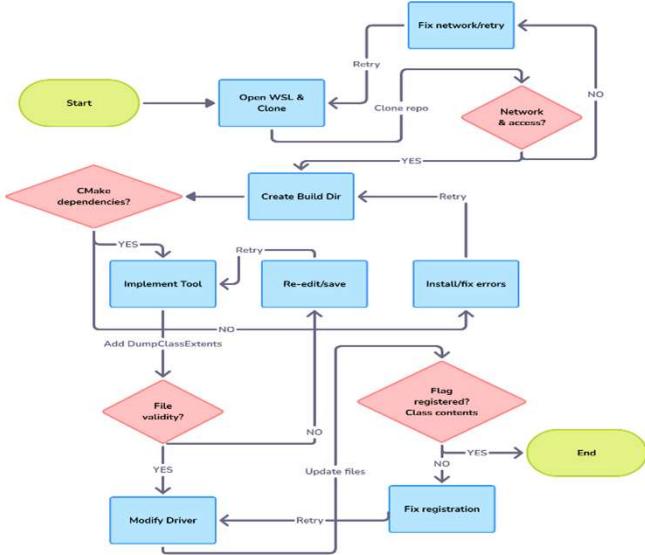
Problem Definition

In modern C++ software systems, knowing the structure and boundaries of class definitions is vital for static analysis, refactoring, and documentation. However, standard compiler outputs lack direct lexical details like file names and line numbers. Developers often resort to external tools, which can be inefficient and error-prone for large codebases. This project fills that gap by extending the Clang compiler with a custom frontend action that extracts and dumps lexical extents of class definitions in a structured format during compilation.

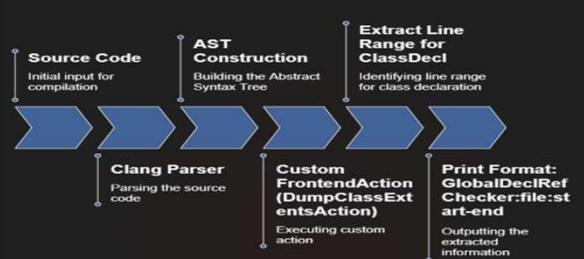
Objectives

- To modify the Clang compiler source to recognize a new command-line option: `-fdump-class-extents`.
- To implement AST traversal logic that detects all class-like constructs including classes, structs, unions, and template classes.
- To extract and print the fully qualified name, source file path, and the starting and ending line numbers (lexical extent) of each class.
- To display the output as: GlobalDeclRefChecker:test.cpp:7-10

Methodology



Compiler Workflow for Class Declaration Extraction



Tools used

Tool	Functionality
LLVM + Clang	Compiler infrastructure
CMake + GNU	Build system
WSL + Bash	Linux environment
Text Editors	Source file editing
Git + G++	Version control and compiling

Results and Discussions

Testing file: test.cpp

```

$ cd /mnt/c/Users/vaibhav/CLionProjects/test
$ g++ -std=c++11 test.cpp
$ ./test
GlobalDeclRefChecker:test.cpp:7-10
GlobalDeclRefChecker:test.cpp:12-16
  
```

Output: WSL(Windows Subsystem for Linux)

```

vaibhav@DESKTOP-ACK9G4N:~$ nano test.cpp
vaibhav@DESKTOP-ACK9G4N:~$ ~/llvm-project/build/bin/clang -cc1 -fdump-class-extents test.cpp
GlobalDeclRefChecker:test.cpp:2-5
GlobalDeclRefChecker:test.cpp:7-10
GlobalDeclRefChecker:test.cpp:12-16
vaibhav@DESKTOP-ACK9G4N:~$ 
  
```

Conclusions

- This project successfully demonstrates the extension of the Clang compiler to extract and dump lexical information of C++ class definitions.
- The seamless integration into the Clang frontend ensures both accuracy and performance.
- The seamless integration into the Clang frontend ensures both accuracy and performance, making it highly suitable for large-scale C++ codebases.

References

- [1] C. Lattner and V. Adve, "LLVM: A Compilation Framework for Lifelong Program Analysis & Transformation," *Proc. CGO*, 2004
- [2] A. Zeller, "Yesterday, My Program Worked. Today, It Does Not. Why?" *Proc. ESEC/FSE*, 1999
- [3] J. Gregor, "Extending Clang for Source-Based Code Analysis," *LLVM Dev Meeting*, 2008.

QR Code(Demonstration Video) and Team Members



Name	USN
Vaibhav U Navalagi	IRV22CS222
Vishwanath A D	IRV22CS234
Wilson	IRV23CS421