Visual Studio Code Verbose Feedback Extension
Student Manual

Dirk Vet
dirk.vet@student.uva.nl

June 10, 2022

Contents

1 Functionality

The Verbose Feedback extension is made for novice programmers to teach them C more quickly and effectively. The extension is able to return errors for the Clang and GCC compilers through various methods. To teach the learners in a efficient manner, the errors from the Clang compiler are delivered together with a verbose description and possibly with a hint to solve the issue. The extension is based on the "C/C++ Advanced Lint for VS Code" extension from Joseph Benden and makes also use of the Microsoft C/C++ extensionBenden n.d.Microsoft n.d.

2 Dependencies

The extension is developed for VScode 1.67.2 on Ubuntu 20.04.4 LTS. Earlier versions of VScode or other operating systems might not be supported. Furthermore, the extension requires a Clang and GCC compiler and the Microsoft C/C++ VScode extension Microsoft n.d. The Clang and GCC compiler has to be installed by the learner, whereas the C/C++ extension can be installed with help from the Verbose Feedback extension.

The Clang compiler can be installed on Ubuntu with: `sudo apt-get install clang`

Dependencies:
VScode: 1.67.2
GCC: 9.4.0
Clang: 10.0.0-4ubuntu1

3 Commands

By using `ctrl` + `P` and typing "Verbose Feedback" you will see a list of available commands as shown below:

- `Verbose Feedback: Switch on/off Clang compiler with feedback.`
  - Switch between dynamic 'C/C++' and '[Clang] Verbose Feedback' diagnostics to turn on/off feedback support.
- `Verbose Feedback: Check for GCC runtime errors/warnings.`

    - Check for runtime errors by compiling all .c files in the current directory using the GCC compiler and by running the executable. Returns static '[GCC] Verbose Feedback' diagnostics.

- Verbose Feedback: Check for Clang runtime errors/warnings.
    - Check for runtime errors by compiling all .c files in the current directory using the Clang compiler and by running the executable. Returns static '[Clang] Verbose Feedback' diagnostics.

- Verbose Feedback: Compile/link the active document with GCC.
    - Check for compiler or linking errors by compiling all .c files in the current directory using the GCC compiler. Returns static '[GCC] Verbose Feedback' diagnostics.

- Verbose Feedback: Compile/link the active document with Clang.
    - Check for compiler or linking errors by compiling all .c files in the current directory using the Clang compiler. Returns static '[Clang] Verbose Feedback' diagnostics.

- Verbose Feedback: Compile and run the program.
    - Compile and run the program.

- Verbose Feedback: Clear all errors by reloading the window.
    - Clears all errors by reloading VScode.

4 Error types

The extension produces four types of error. These differ in the way that they appear in the Problems window.

- Static diagnostics: diagnostics appear and disappear by executing a command through ctrl+P. These are prefixed with *[S]*.

- Dynamic diagnostics: diagnostics appear and disappear while typing. These are prefixed with *[D]*.

4.1 Dynamic ’C/C++’ errors

Errors with the label ’C/C++’ are generated by the Microsoft C/C++ extension without invoking a specific compiler. These are only visible when the Verbose Feedback mode is disabled and appear/disappear while typing. To switch the Verbose Feedback mode on or off you should use the Verbose Feedback: Switch on/off Clang compiler with feedback. command.

4.2 Static ’gcc’ errors

In case of using the Verbose Feedback: Compile and run the program. command VS-code will ask on the first call for which compiler you want to use. This results in VScode creating a `tasks.json` file in `./.vscode`.

The errors generated by the compiler will be shown in the Terminal window under the C/C++: gcc build active file tab. Most of these errors will be shown in the Problems window with the label ’gcc’. These are static and only disappear if you run the Verbose Feedback: Compile and run the program. command again.

4.3 Dynamic ’[Clang] Verbose Feedback’ errors

These errors are generated by the Clang compiler and contain besides the original Clang error the phase of the compiler at which the error occurs, a simplified description of the error and possibly a hint. The errors appear/disappear while typing. The Verbose Feedback mode can be activated with the Verbose Feedback: Switch on/off Clang compiler with feedback. command. Once active the C/C++ errors disappear from the Problems window.

4.4 Static ’[<*Compiler*>] Verbose Feedback’ errors

These diagnostics always contain the error location [Ln 1, Col 1]. It is not sure if the error message is a true error or a print statement to stderr from the program, so no proper error location can be given. For the true error location you should read it directly from the original error message.

These error messages have labels with the prefix [GCC] or [Clang] depending on the compiler that has been used. The diagnostics disappear when a command is run without problems.

5 Run program

The extension is able to run a program as well with the Verbose Feedback: Compile and run the program. command, which displays output in the terminal. On the first time you will need to select a compiler which you have installed on your machine, which will generate a `tasks.json` file in `./.vscode`. You might want to change the "tasks.args" field to add compiler flags, additional files that have to be compiled and arguments. Parameters for the compile command for tasks.json that includes all .c files in the current directory could be for example:

```
"args": [
"-fdiagnostics-color=always",
"-g",
"${fileDirname}/*.c",
"-o",
"${fileDirname}/${fileBasenameNoExtension}"
]
```

In case you want to use a different compiler press F5 and select C++ (GDB/LLDB) followed by your preferred compiler. This which will run and debug your file, and will add a configuration in `./.vscode/tasks.json`. You might have to change the "tasks.args" field again depending on your needs. When using Verbose Feedback: Compile and run the program. again, keep in mind which compiler it will use. To use GCC make sure that the preferred GCC configuration in the JSON file contains `"tasks.group.isDefault" : true`. Furthermore, all the compiler errors will be shown with the 'gcc' label, regardless of the compiler being used. These errors will only disappear when the program is compiled and run again.

After compiling successfully, when the program has to be ended prematurely use ctrl+C in the Terminal window, ⇧+F5, or the Stop button.

6 Tips

- There is a choice between GCC and Clang. Even though your final program will probably be run by one specific compiler, use both compilers during code development as both compilers give different error messages.

- Note that all commands that result in static errors compile all the .c files in your current directory. This means you can only have one main() function in each directory.

- The dynamic 'C/C++' and static 'gcc' diagnostics do not contain any [D] or [S] label respectively.
- Errors and command containing the label 'C/C++' are related to the Microsoft C/C++ extension. A lot of support for this can be found online, which will also be applicable to the Verbose Feedback extension.

References

Benden, Joseph (n.d.). *C/C++ Advanced Lint for VS Code*. Retrieved May 27 2022. URL: `https://github.com/jbenden/vscode-c-cpp-flylint`.

Microsoft (n.d.). *C/C++ for Visual Studio Code*. Retrieved May 27 2022. URL: `https://marketplace.visualstudio.com/items?itemName=ms-vscode.cpptools`.