IDEs and debugging JEDI

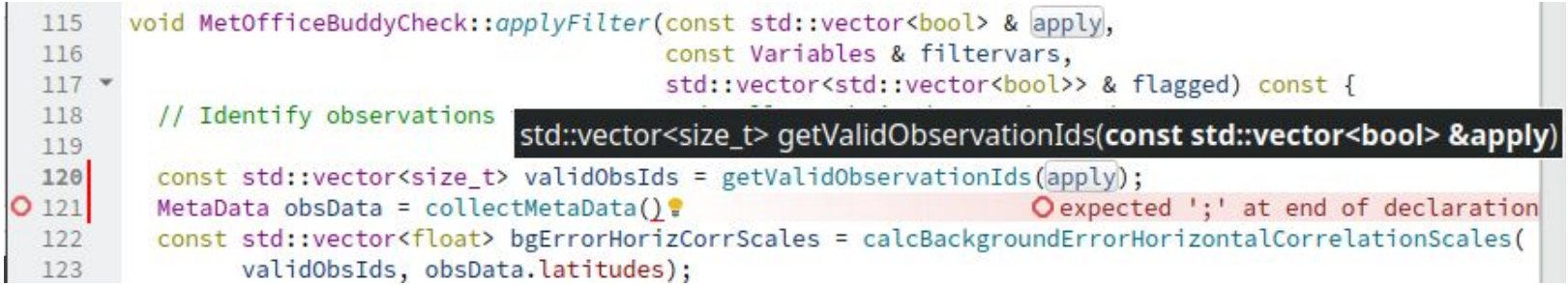
Wojciech Smigaj (Met Office)

Some popular IDEs with C++ support

- CLion
- Eclipse
- Qt Creator
- Visual Studio Community/Professional/Enterprise (Windows only)
- Visual Studio Code
- XCode (Mac only)

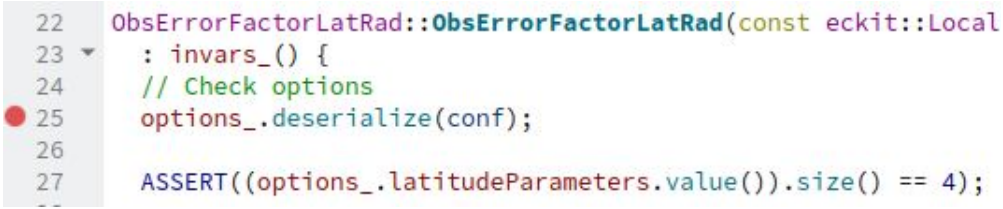
IDE benefits

- Quick code navigation
- Autocompletion, contextual help
- On-the fly detection of syntax errors
- Help with refactoring (e.g. renaming variables)
- Integrated debugger



Debugger basics

- **Breakpoints**: code locations where the debugger will interrupt a program. Can be unconditional or conditional.



- Once a breakpoint is hit, the developer can examine the state of the program (e.g. inspect the **call stack** and values of **local variables**).
- Subsequently, they can resume normal execution or **step through** the code.
  - **Step over**: execute a line of code as a whole.
  - **Step into**: like *Step over* unless the current line contains a function call. If so, suspend execution at the first statement in that function.
  - **Step out**: execute code until the current function returns.

Debugging: compilation flags

Binaries must contain (or be accompanied by) debugging symbols.

- To build JEDI with debugging symbols, select the *Debug* or *RelWithDebInfo* configuration with CMake/Ecbuild
  - ecbuild --build=Debug *path_to_source_folder*
  - cmake -DCMAKE_BUILD_TYPE=Debug *path_to_source_folder*
- Compiling with optimisations makes debugging harder. If you can afford it, use the Debug rather than the RelWithDebInfo configuration.