



CMakeLists.txt

```
#The root CMakeLists.txt file.
cmake_minimum_required(VERSION 3.28)

set(CMAKE_TOOLCHAIN_FILE "$ENV{VCPKG_ROOT}/scripts/buildsystems/vcpkg.cmake" CACHE STRING "Vcpkg toolchain file")

#The project name is set here. In this case it's Rocket, but you can change it to whatever you want.
project(Rocket VERSION 1.0.0 LANGUAGES CXX)

#Require C++23
set(CXX_STANDARD_REQUIRED ON) #Make C++23 a hard requirement
set(CMAKE_CXX_STANDARD 23) # Default C++ standard for targets
set(CMAKE_CXX_SCAN_FOR_MODULES ON)

find_package(fmt CONFIG REQUIRED)

#We're using the project name as the target name, but you can change it to make them different.
#With this setup, the name of the executable will be the same as the project name.
add_executable(${PROJECT_NAME} main.cpp)

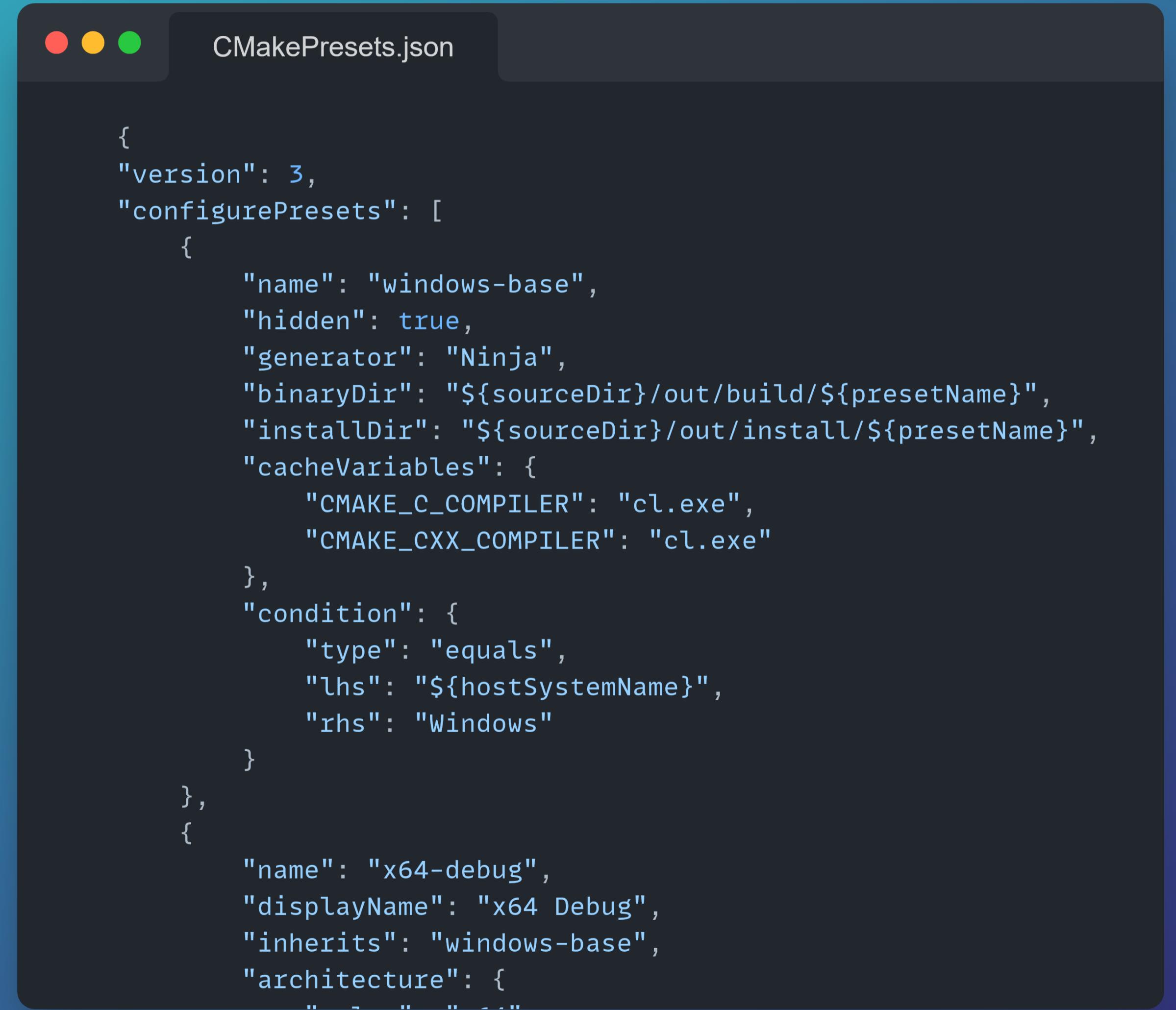
target_sources(${PROJECT_NAME}
    PUBLIC
        FILE_SET CXX_MODULES FILES
            utilities.ixx
)
target_link_libraries(${PROJECT_NAME} PRIVATE fmt::fmt)
```



vcpkg.json

```
{
    "dependencies": [
        "fmt"
    ]
}
```

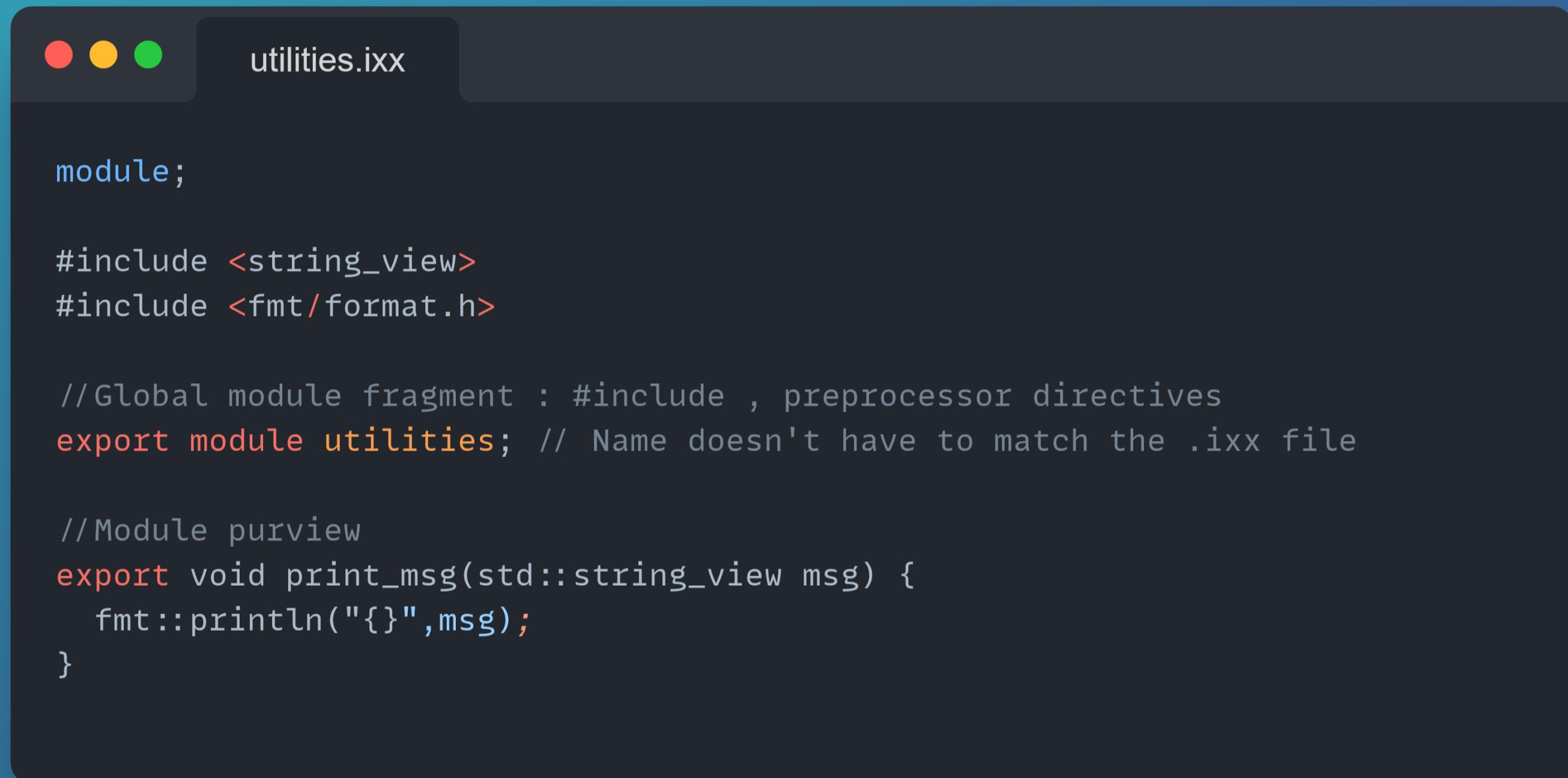
The CMake presets file



A screenshot of a dark-themed code editor window titled "CMakePresets.json". The file content is a JSON object representing CMake presets. It includes configuration for a "windows-base" preset (hidden, Ninja generator, specific binary and install dirs, and cache variables for compilers) and an "x64-debug" preset (inherits from windows-base, specific architecture).

```
{
  "version": 3,
  "configurePresets": [
    {
      "name": "windows-base",
      "hidden": true,
      "generator": "Ninja",
      "binaryDir": "${sourceDir}/out/build/${presetName}",
      "installDir": "${sourceDir}/out/install/${presetName}",
      "cacheVariables": {
        "CMAKE_C_COMPILER": "cl.exe",
        "CMAKE_CXX_COMPILER": "cl.exe"
      },
      "condition": {
        "type": "equals",
        "lhs": "${hostSystemName}",
        "rhs": "Windows"
      }
    },
    {
      "name": "x64-debug",
      "displayName": "x64 Debug",
      "inherits": "windows-base",
      "architecture": {
        "id": "x64"
      }
    }
  ]
}
```

The module file



A screenshot of a terminal window titled "utilities.ixx". The window has a dark background and light-colored text. It contains the following code:

```
module;

#include <string_view>
#include <fmt/format.h>

// Global module fragment : #include , preprocessor directives
export module utilities; // Name doesn't have to match the .icxx file

// Module purview
export void print_msg(std::string_view msg) {
    fmt::println("{}" ,msg);
}
```

The main.cpp file



A screenshot of a terminal window titled "main.cpp". The window has three standard Mac OS X window controls (red, yellow, green) at the top left. The title bar contains the text "main.cpp" next to a small C++ icon. The terminal window displays the following code:

```
// One line comment
import utilities;

int main() {
    print_msg("Hello world!");
}
```

CMake: CMakeLists.txt and CMakePresets.json

Vcpkg: vcpkg.json

C++ compiler: main.cpp and utilities.ixx

Environment setup flow

