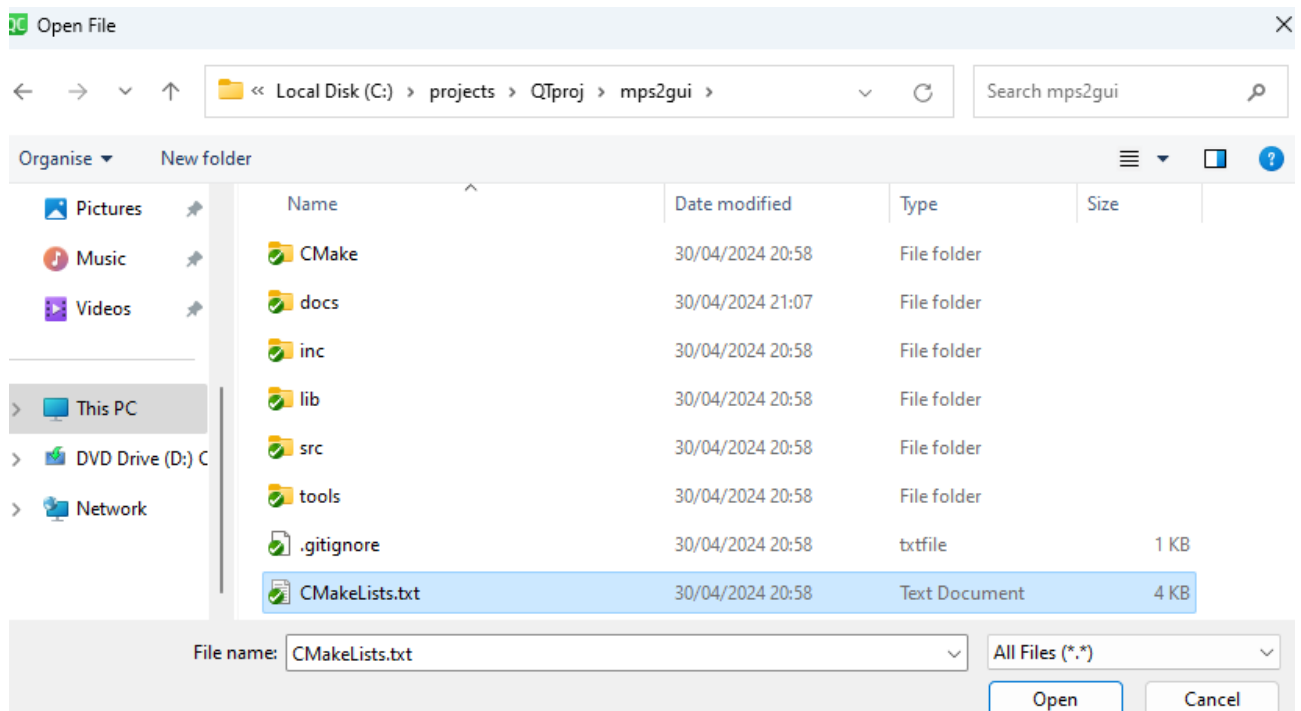
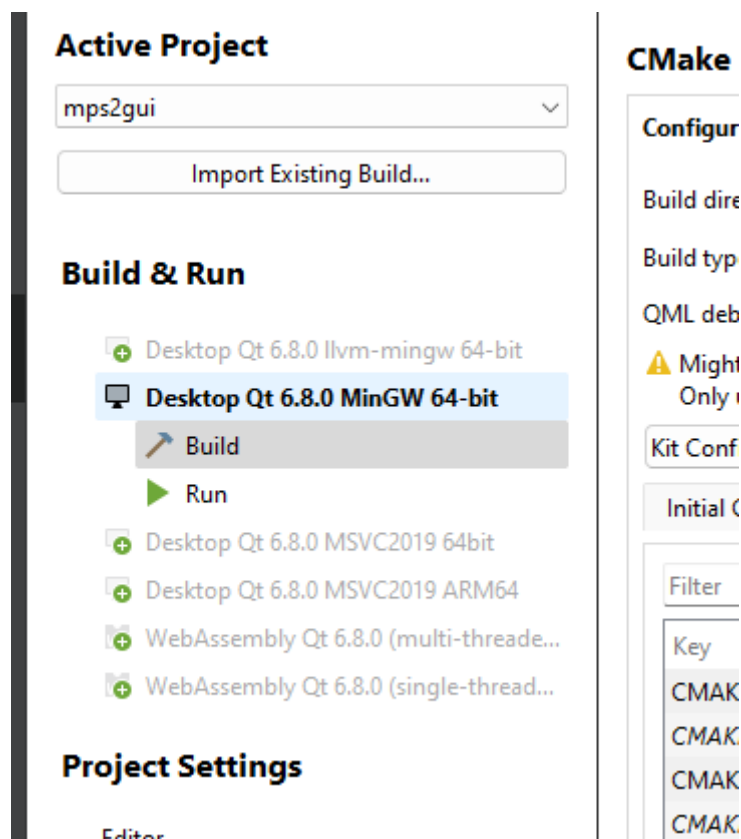
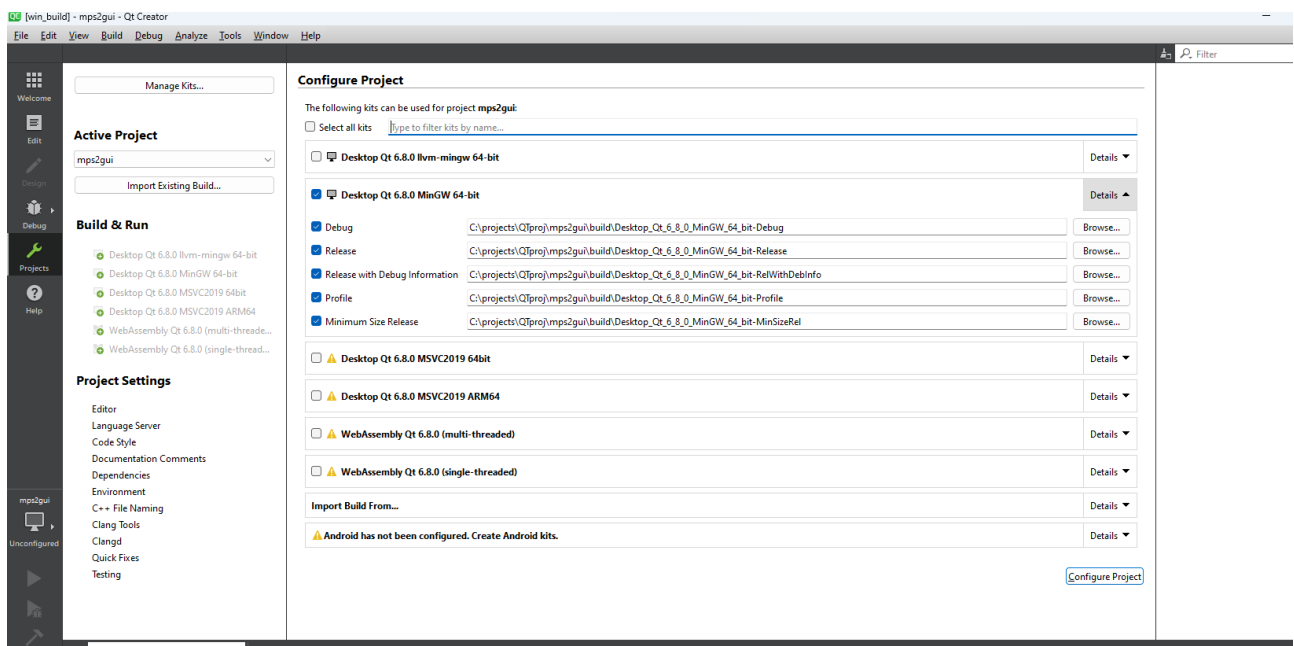


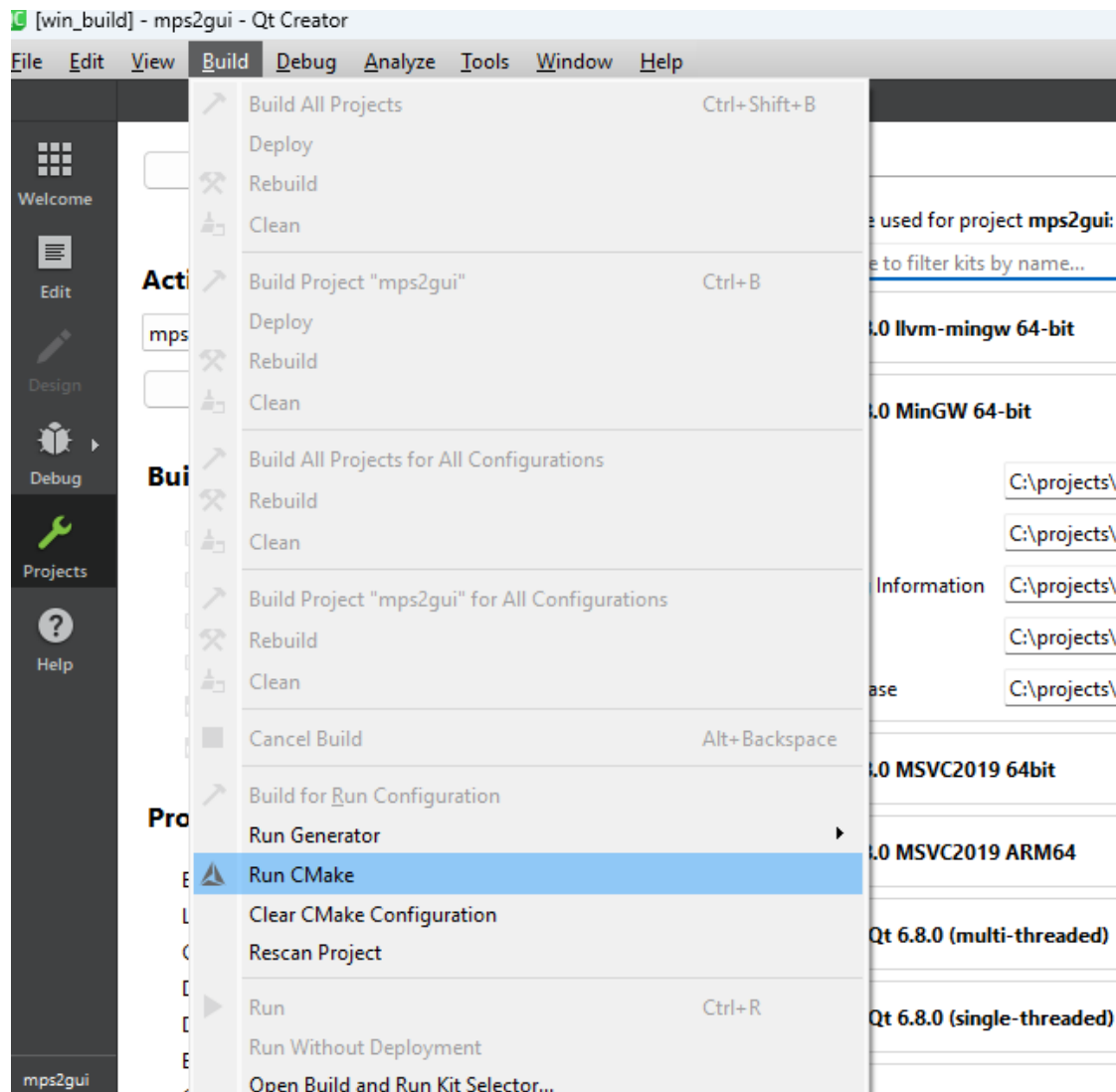
-
- Qt Creator
- File Edit View Build Debug Analyze Tools Window Help
- Welcome
- QC Welcome to Qt Creator
- Create Project...
- Open Project...
- Qt6 6.8.0
- Search in Examples...
- Application Examples
- Calqlatr
- Tags: quick
- Car Configurator
- Tags: 3d camera qui



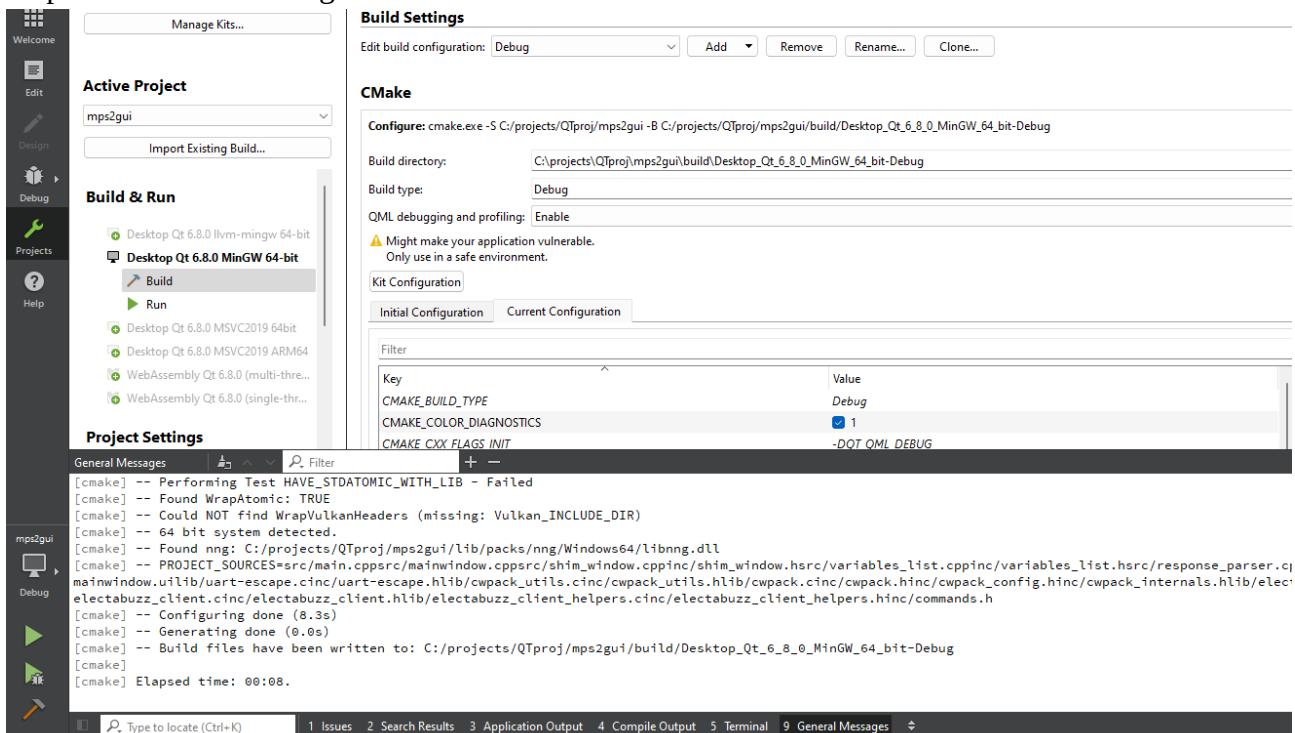
3. Use MINGW option (default are ok)



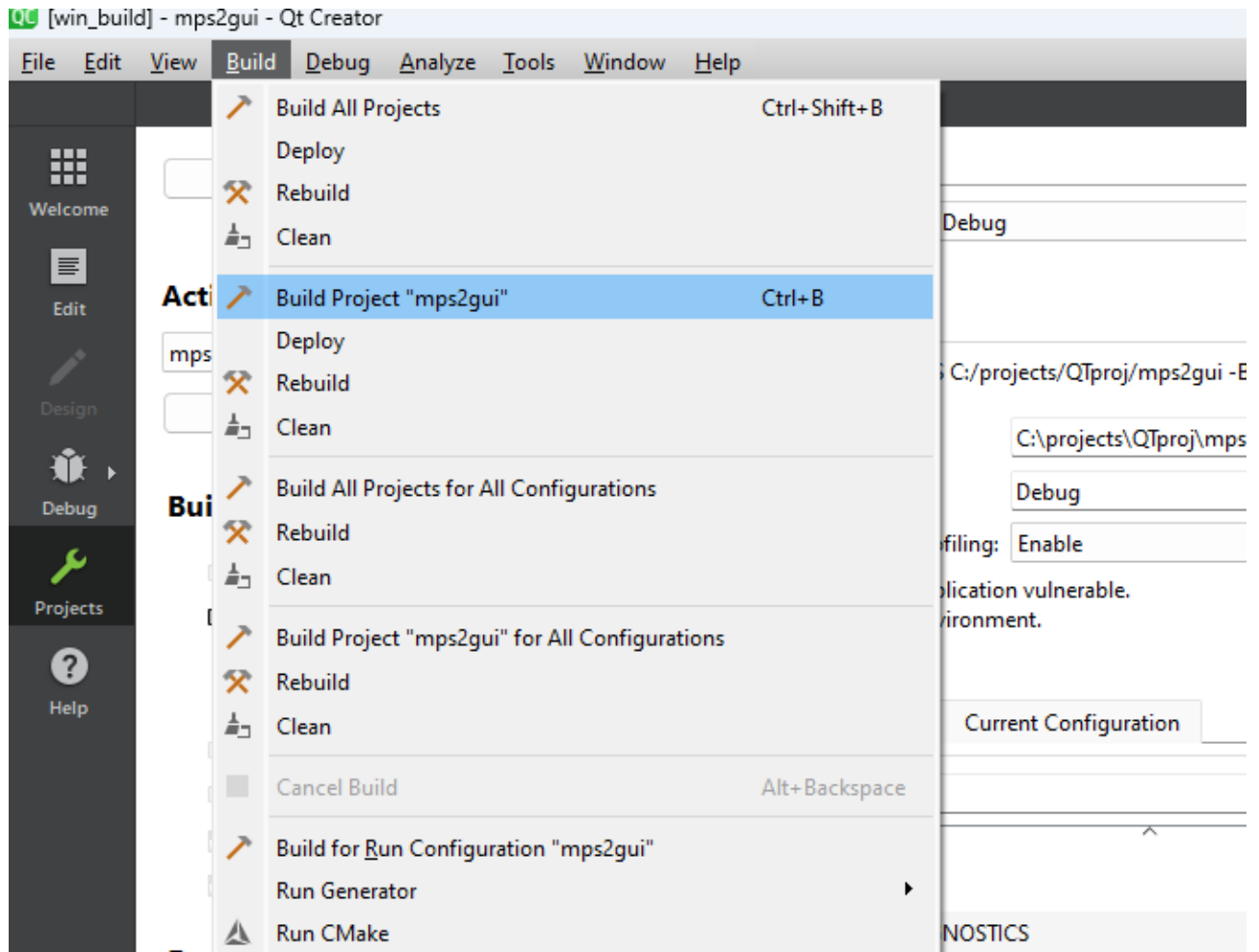
4. Go Build – Run Cmake



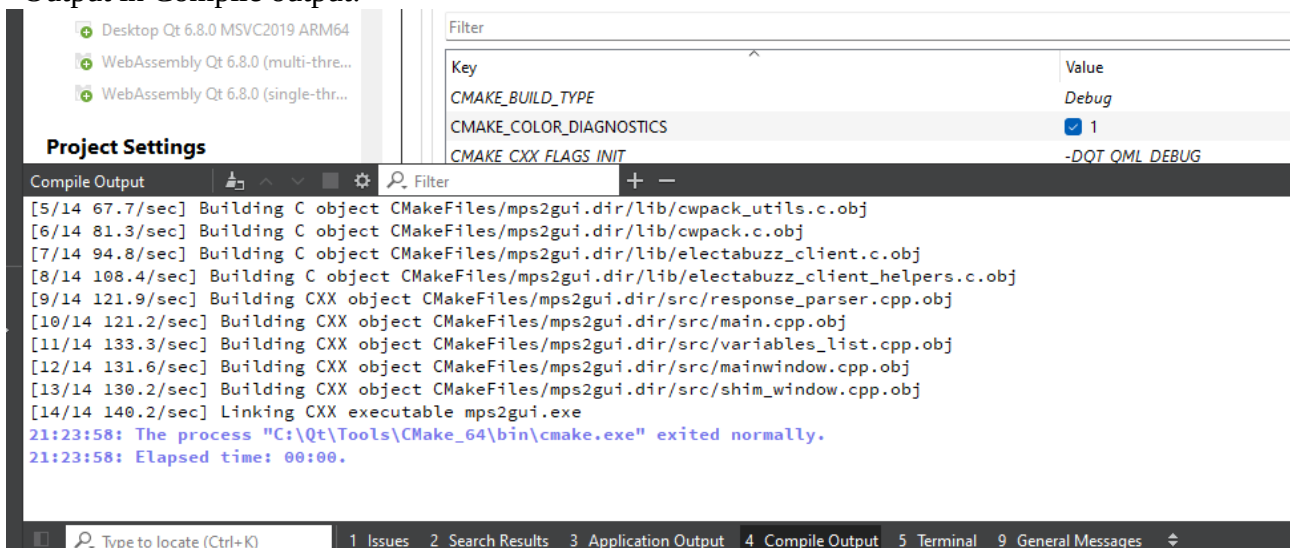
Output in General Messages:



5. Go Build – Build Project

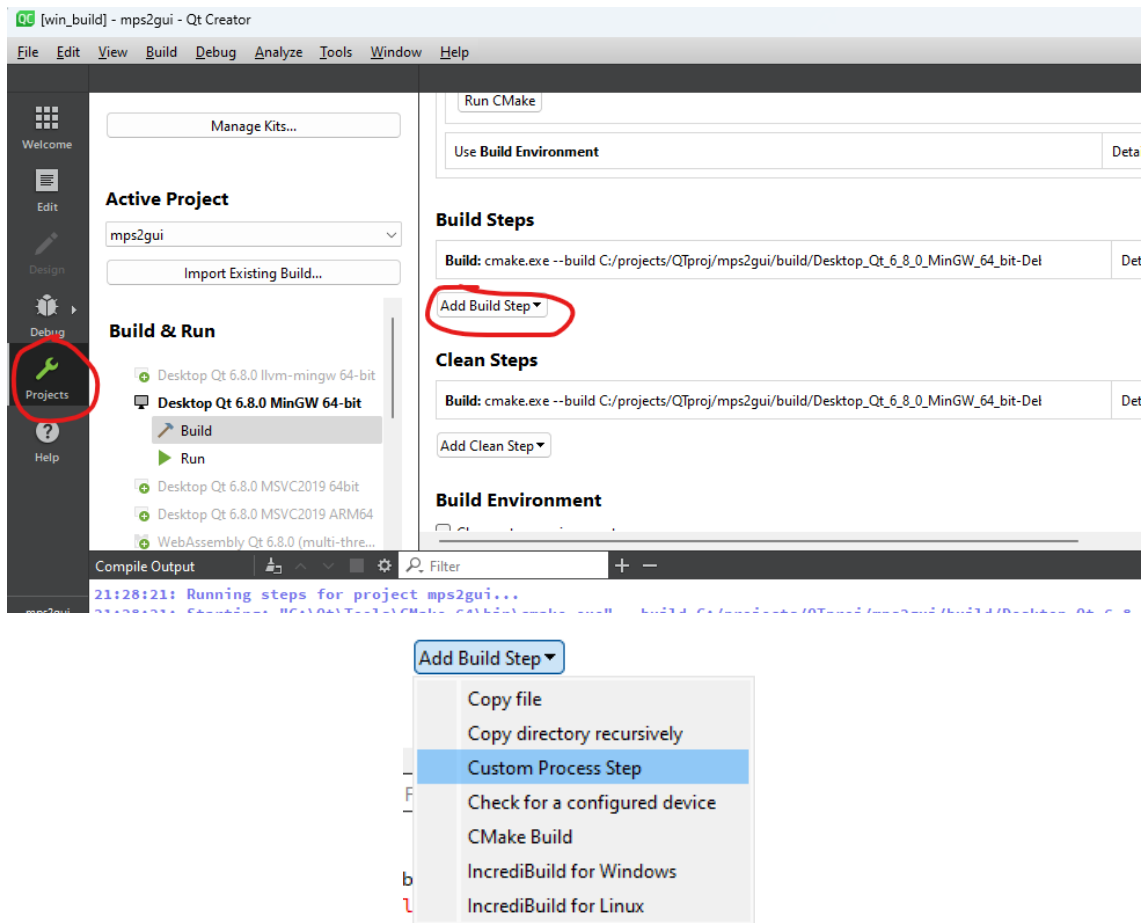


Output in Compile output:



At this stage project can start by pressing “Run” from Qt Creator.
Prepare executable in next step.

6. Go to Project settings and to Build Steps section, press Add Build Step - Custom process step



Fill as follows:

Command:

`%{sourceDir}\tools\post_build.bat`

Arguments:

`%{ActiveProject::QT_HOST_LIBEXECS}\windeployqt.exe %
 {ActiveProject:RunConfig:Executable:NativeFilePath} %
 {ActiveProject:RunConfig:Executable:NativePath}`

Working dir:

`%{ActiveProject:RunConfig:Executable:NativePath}`

Build Steps

Build: <code>cmake.exe --build C:/projects/QTproj/mps2gui/build/Desktop_Qt_6_8_0_MinGW_6</code>	Details
Custom Process Step: <code>post_build.bat C:\Qt\6.8.0\mingw_64\bin\windeployqt.exe C:\pro</code>	Details
Command: <code>%{sourceDir}\tools\post_build.bat</code>	Browse
Arguments: <code>iveProject:RunConfig:Executable:NativeFilePath} %{ActiveProject:RunConfig:Executable:NativeP</code>	
Working directory: <code>%{ActiveProject:RunConfig:Executable:NativePath}</code>	Browse
Add Build Step	

Now in working build directory will be created EXE folder, that contains all nested libs and files.

