


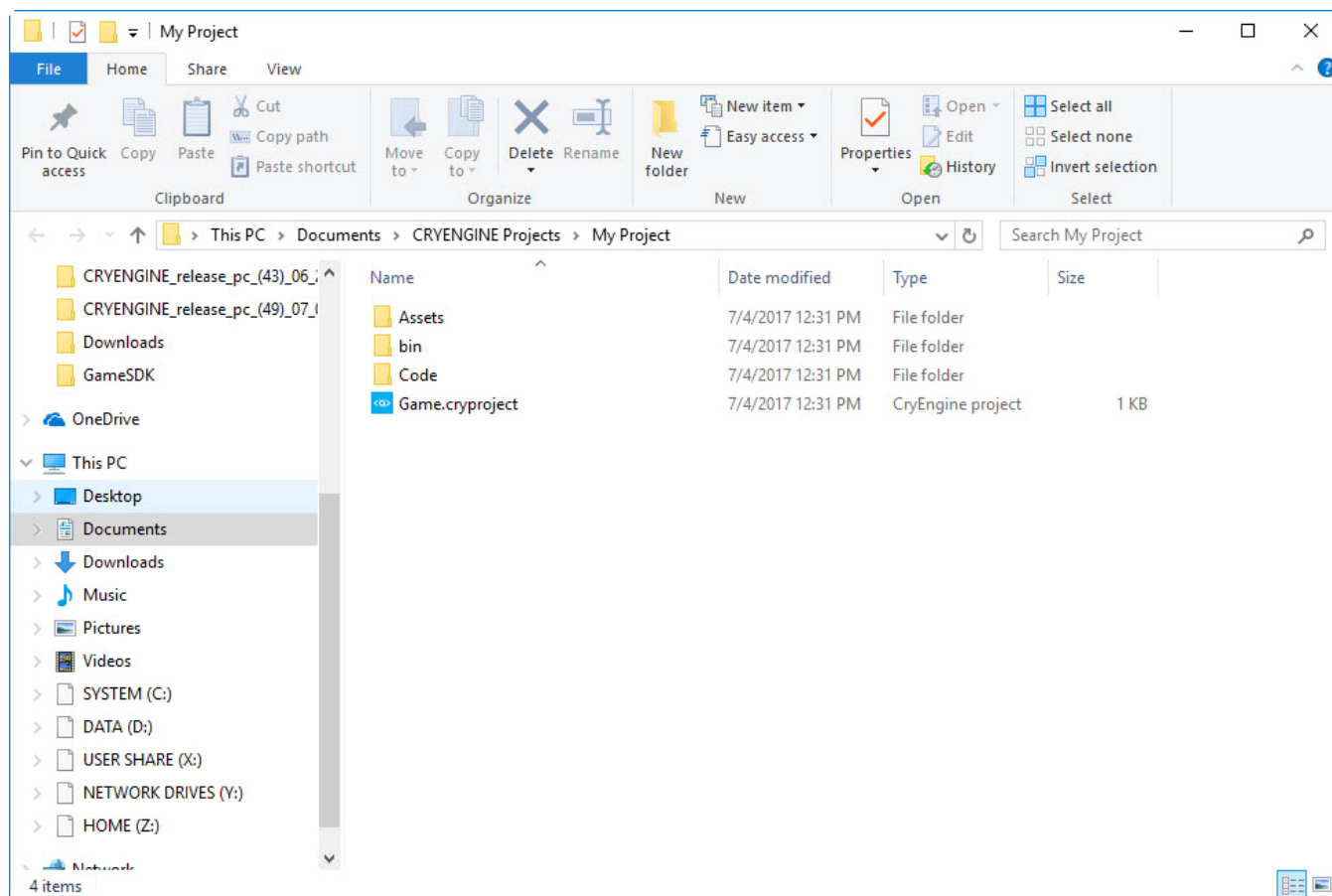
The steps shown below show you how to configure your project to include the engine project inside your project solution file in Visual Studio, which also allows you to build the engine alongside your project.

- [Overview](#)
- [Prerequisites](#)
- [Associating a Project with a Custom Engine](#)
- [Enabling Engine Compilation for a Project](#)

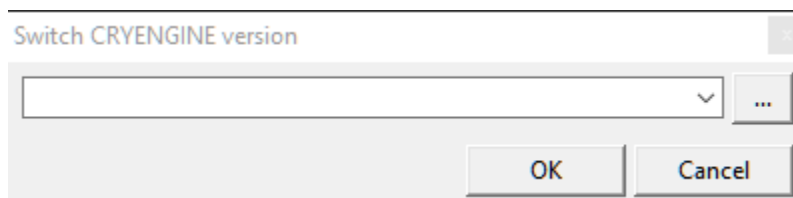
This article assumes that you have already followed the [Building the Engine from Source Code](#) guide that walks you through the process of setting up your local CRYENGINE build via the source code from the CRYENGINE GitHub repository, as well as having [created a project](#) using the [same version](#) of the engine.

 If the associated engine version of the project between the original engine and custom engine differs, or, there have been significant changes to the code base of the custom engine, your project may fail to compile.

To start, we'll need to navigate to the directory in which our `.cryproject` file resides. Once you've done that you should see something similar to the screenshot below:

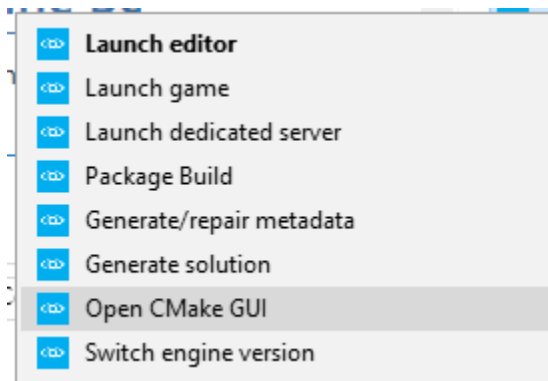


The first step to associate this project with our custom Engine is to right-click the project file and select **Switch Engine Version**. This should bring up the window below:

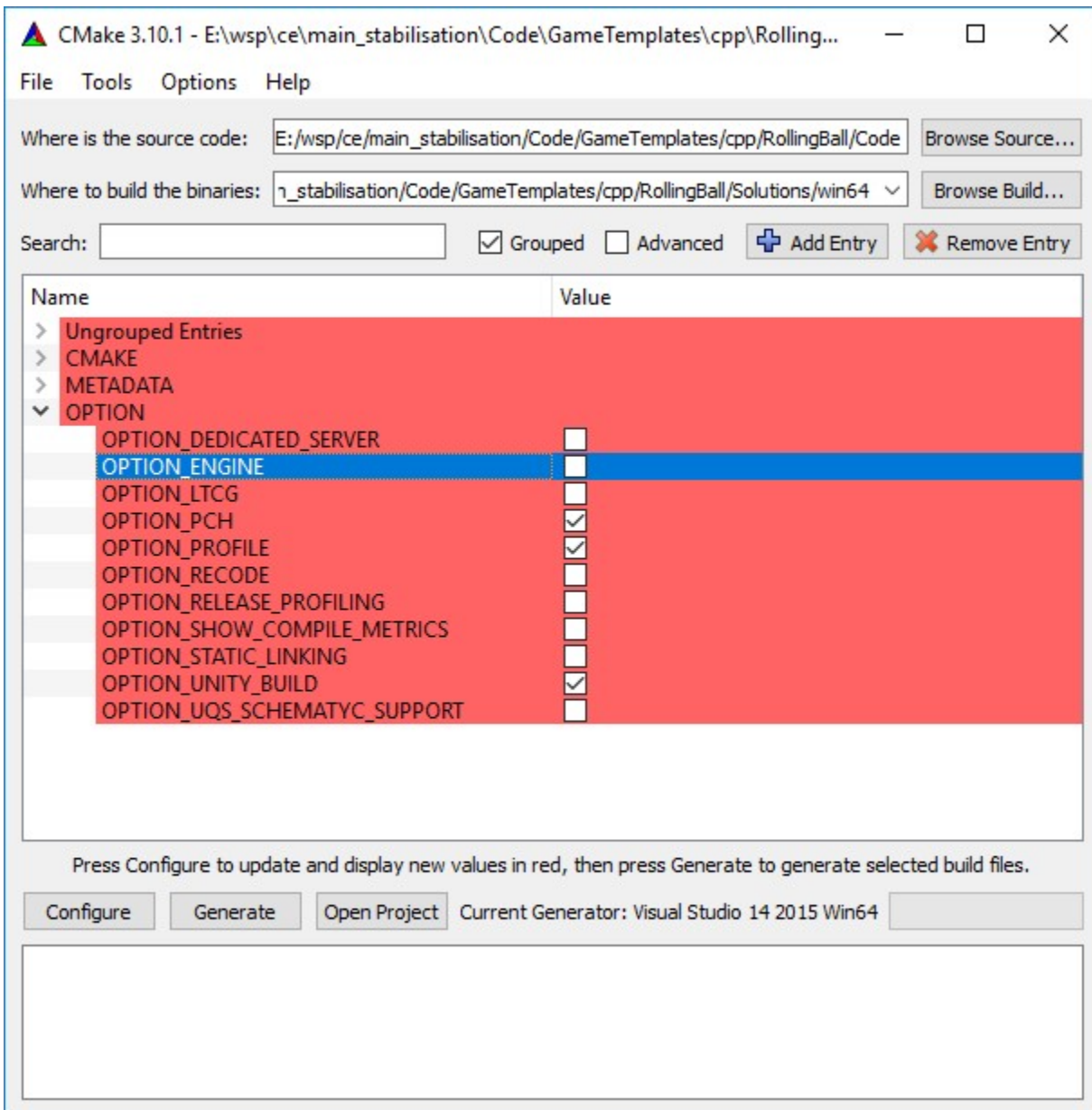


From here you should see the name of the custom engine that we registered in the previous guide in the Drop-Down menu.

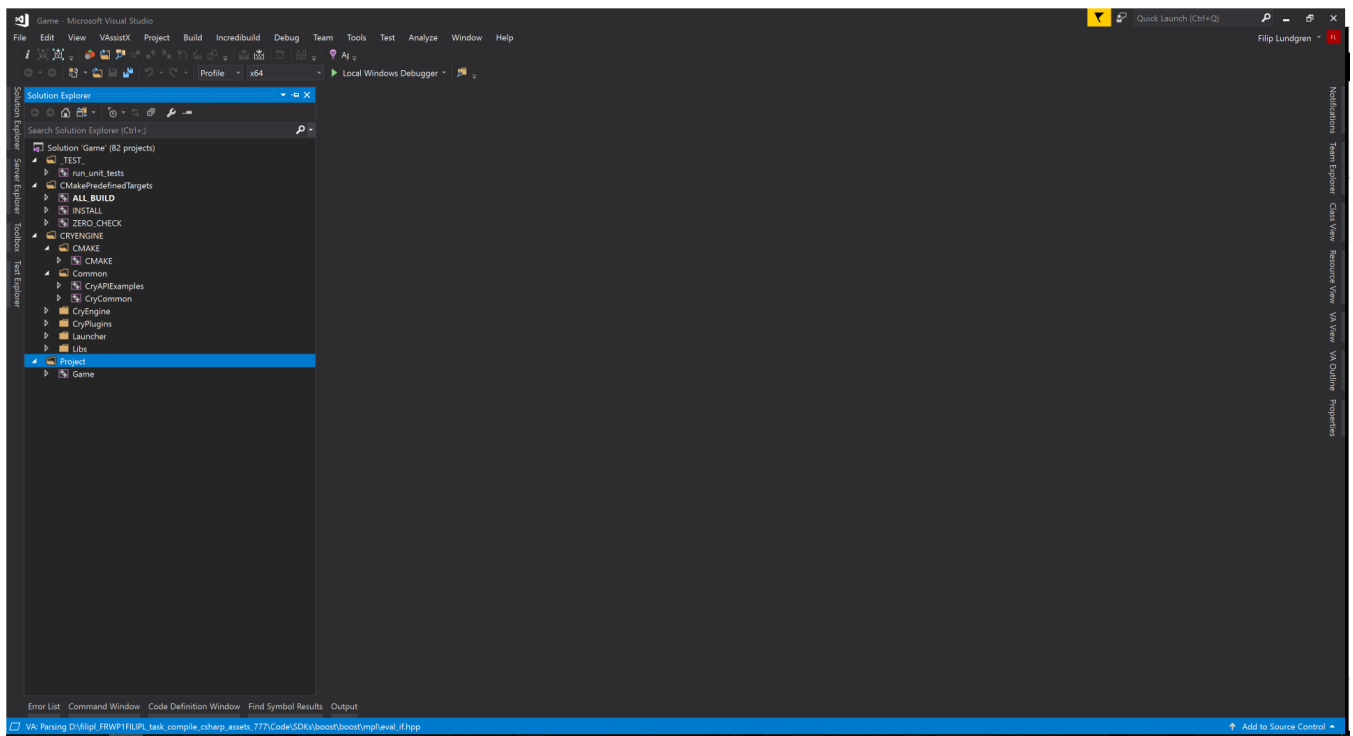
Once a project is associated with the engine we want to build, right-click your `.cryproject` file and select **Open CMake GUI** from the menu:



This will open up the CMake GUI with our project selected. Once opened, tick **OPTION_ENGINE** to enable the engine to be built:



Then press **Generate** to re-generate your solution. We can then click **Open Project** to open the project in Visual Studio, giving us one solution with both engine and game project inside:



Building the project will now compile the engine **and** your project. Note that this will take considerably more time. Engine binaries will now be output to the project directory instead. After this is done we recommend moving the *Engine/*, *Editor/* and *Tools/* folders into your project directory - this ensures that the engine will find all relevant tools during development.