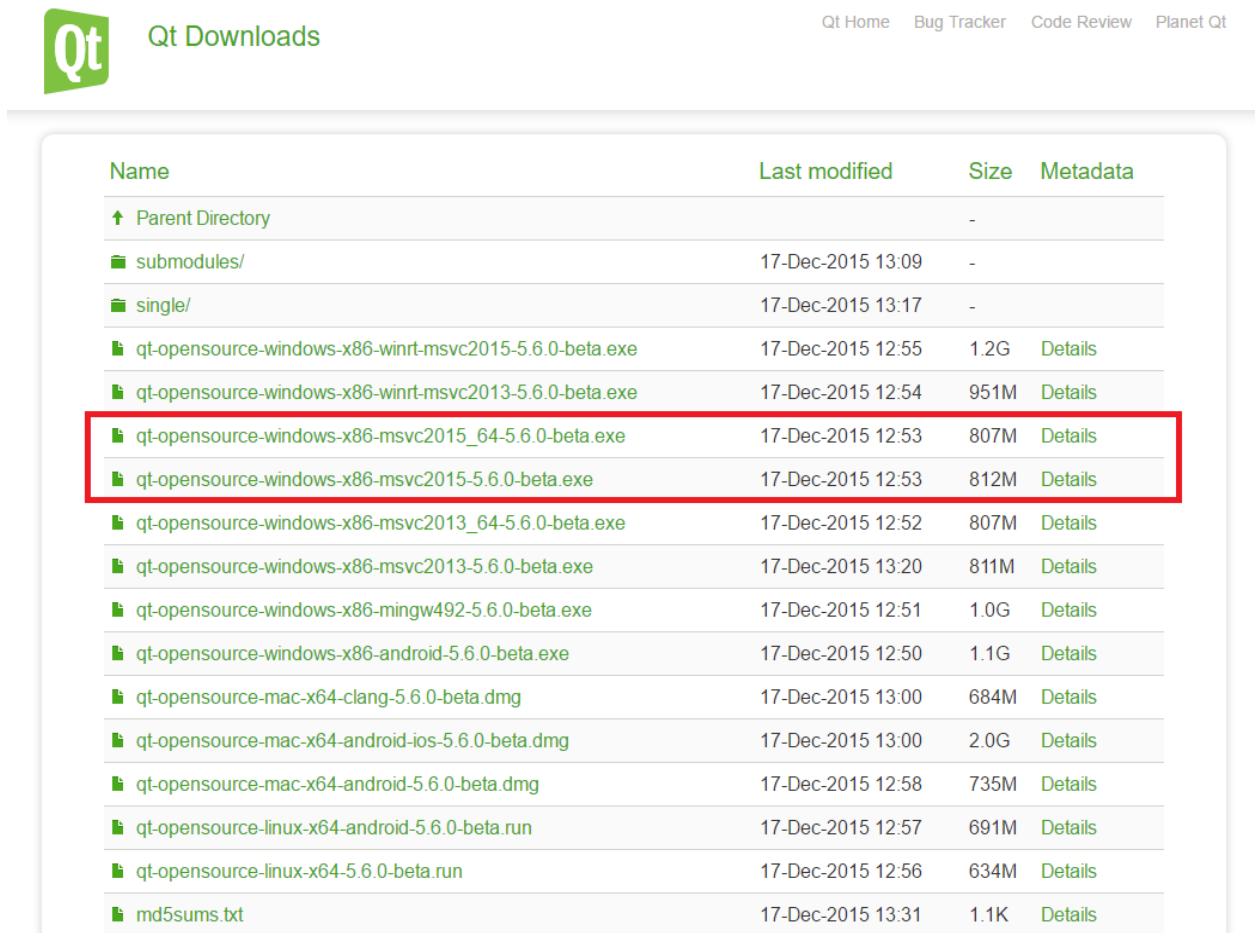# SETTING UP QT WITH VISUAL STUDIO (VS) 2015

1. Go to: https://download.qt.io/development_releases/qt/5.6/5.6.0-beta/.

2. Download one of the following two: *qt-opensource-windows-x86-msvc2015_64-5.6.0-beta.exe* or *qt-opensource-windows-x86-msvc2015-5.6.0-beta.exe*.   Take   "qt-opensource-windows-x86-msvc2015_64-5.6.0-beta.exe" if you have a 64-bit Windows operating system, or the other one, if you have a 32-bit Windows operating system.


3. Install the downloaded file (go with the default settings, there is no need to change anything). When asked for a username, you have the option to skip that step.

4. Open Visual Studio.

5. In the menu, go at "Tools" -> "Extensions and Updates".

6. In the Extensions and Updates window, on the left-hand side column, extend "Online", then "Visual Studio Gallery".

7. In the right-hand side column, in the search window at the top, write "QT5".
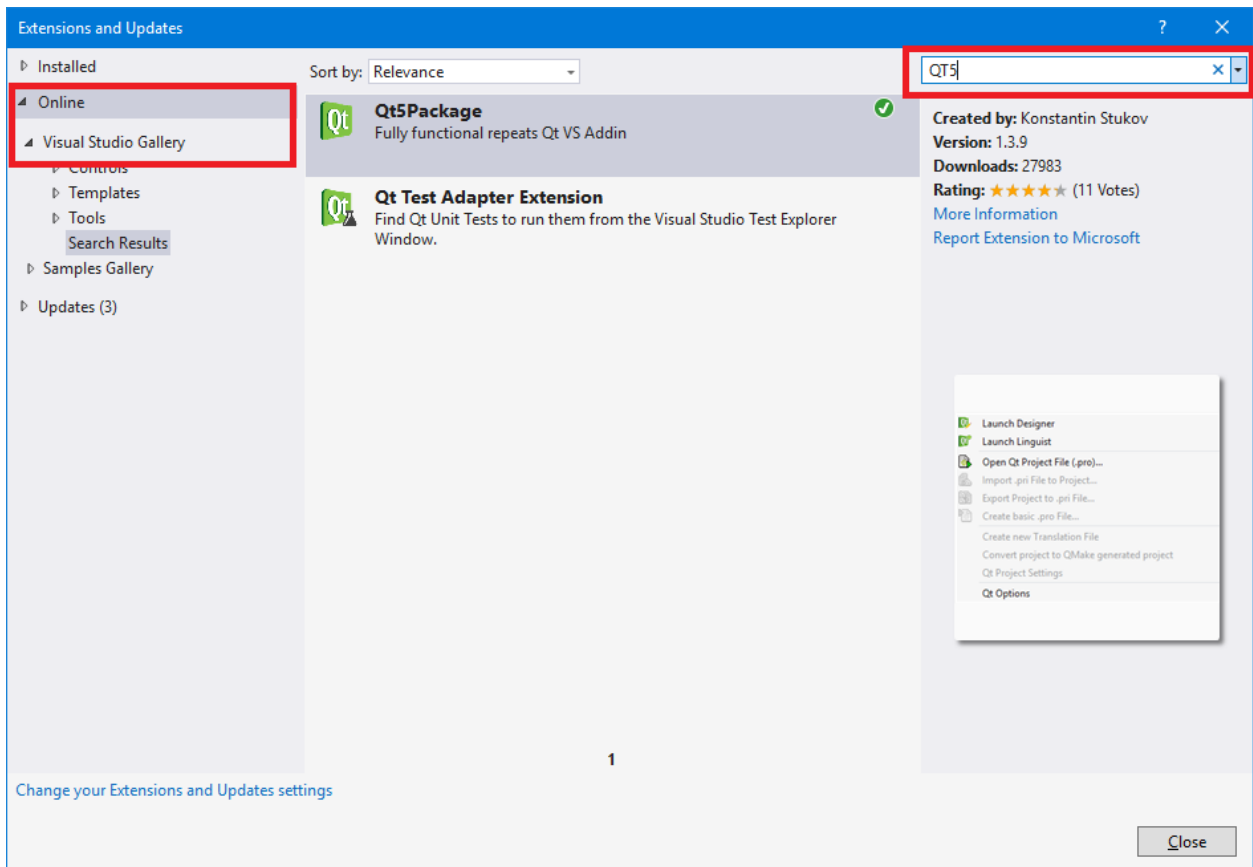
8. The "Qt5Package" should be found and you will see a green "Download button" (in the photo, the button is not there, as I have already installed it ☺). Press the button to download the add-in.

9. After the download finishes, you should have a new item in the VS menu, called "QT5".
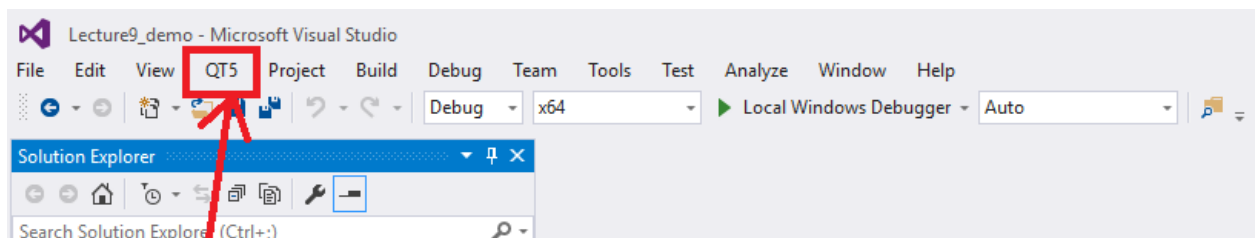
10. Go to "QT5" -> "Qt Options" and in the window press "Add". Fill the new window with the following information:

- Version name: **msvc2015_64** (or, without 64, if needed)
- Path: the path towards the msvc2015_64 folder on your system. In my case, this path is: **c:\Qt\Qt5.6.0\5.6\msvc2015_64**.

Your Qt Options window should now look similar to this:
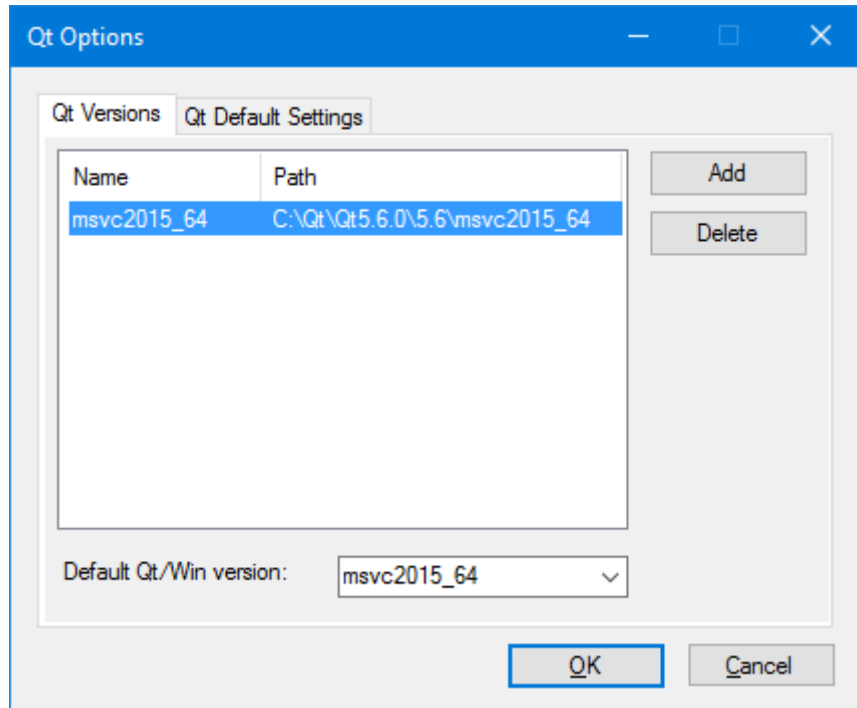
11. Done! Restart Visual Studio. Now, let's see how we can create a new Qt project.

CREATING A NEW QT PROJECT WITH VISUAL STUDIO 2015

1. From the VS menu, select "File" -> "New" -> "Project".

2. In the left-hand column of the "New Project" window, extend "Templates" -> "Visual C++" and select "Qt5 Projects". In the middle column, select "Qt Application". Add a name and a location and then press "OK".
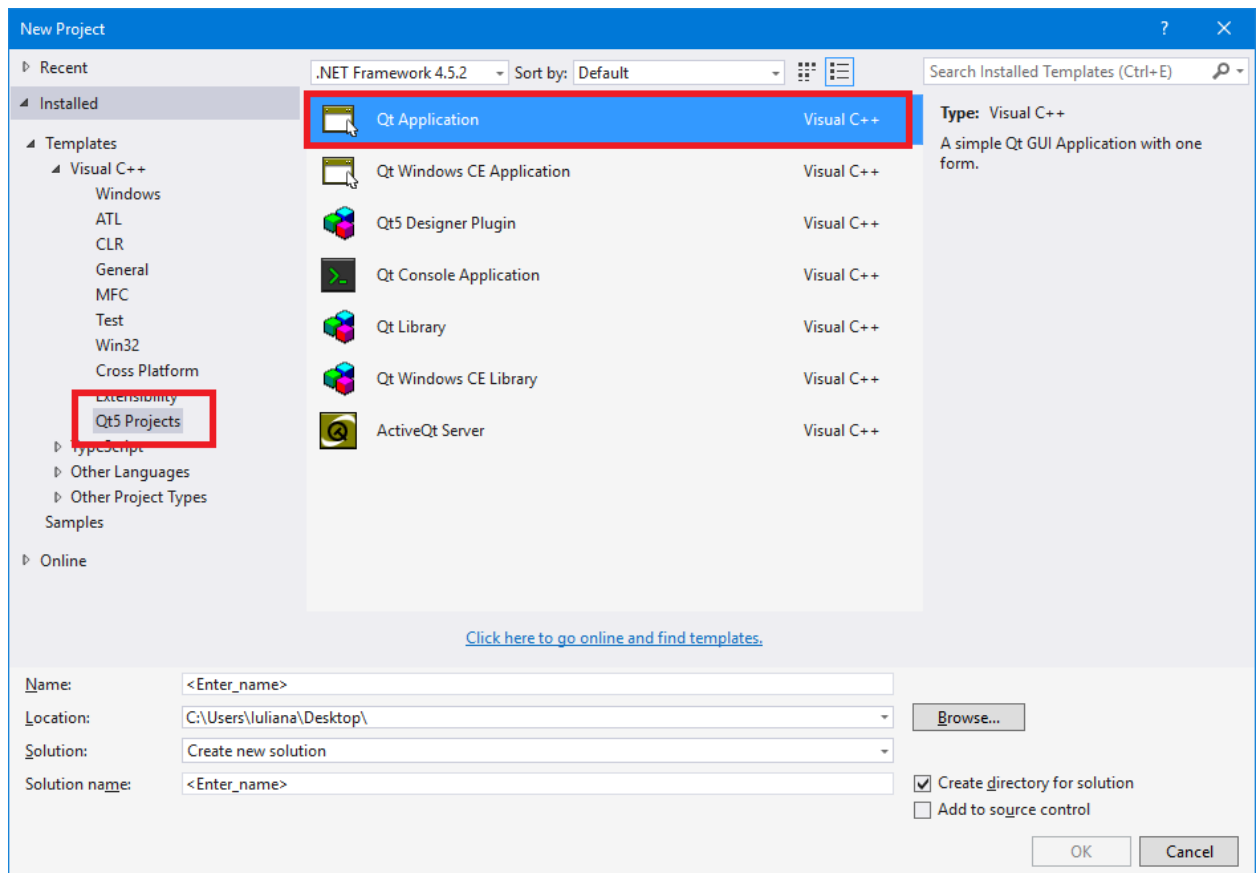
3. In the following dialogs, keep the default settings (Next -> Next -> Finish).

4. In the "Source Files", a file "main.cpp" should already be created.

5. Ignore the red underlines. Run the application. You should see an empty window:
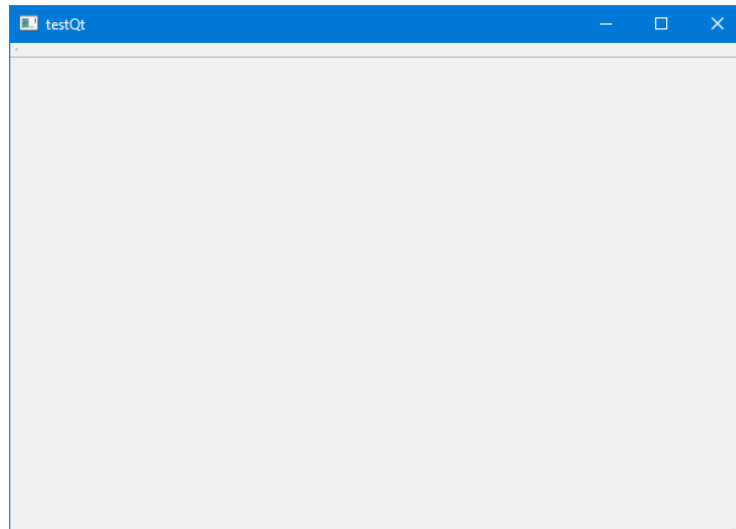
6. The red underlines are not actually compilation errors, they look like that because a VS Intellisense does not detect Qt variables. To fix this, do the following steps.

7. Right-click on the project (in Solution Explorer) -> select "Properties".

8. In the "Property Pages" window, go to "VC++ Directories". Edit the "Include Directories".
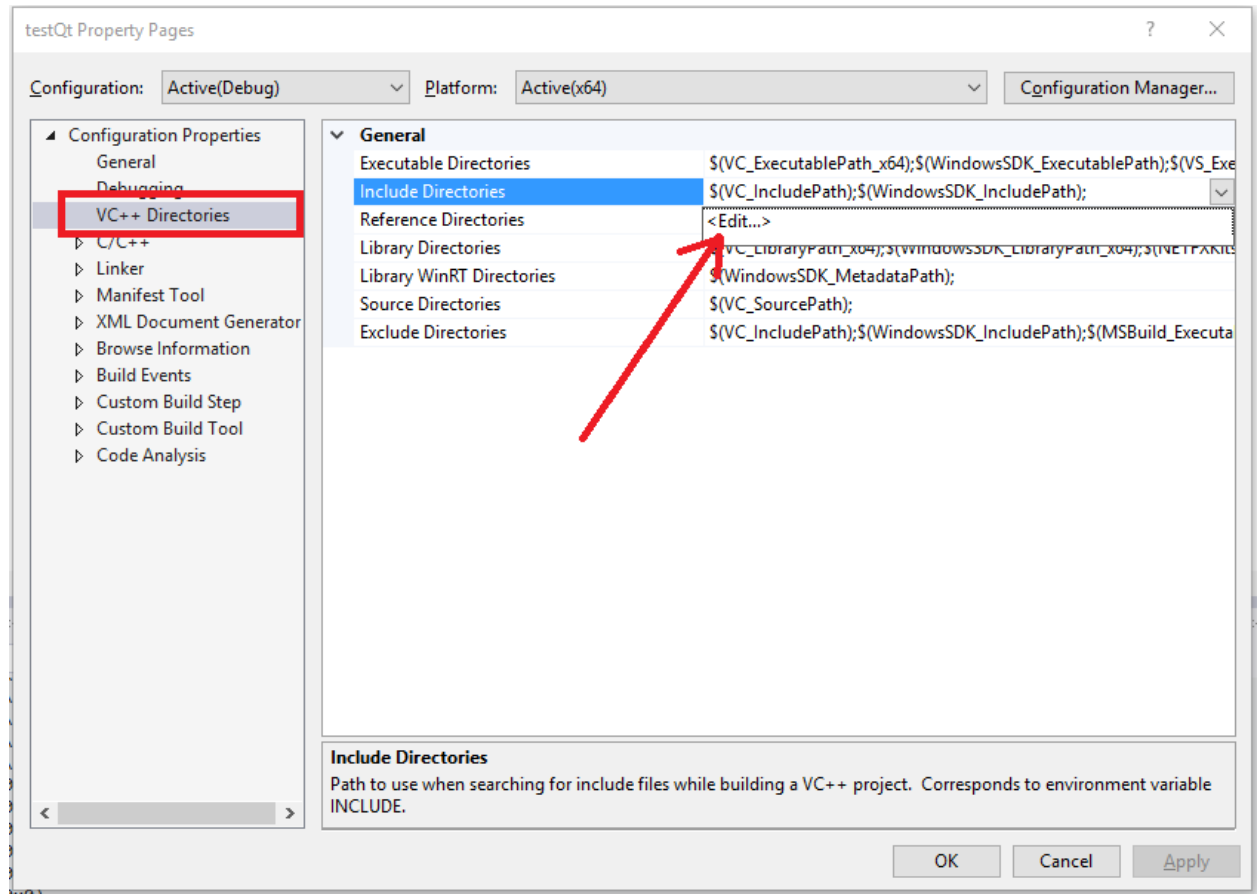
9. Go to the "New line" button.

Include Directories

Evaluated value:

C:\Program Files (x86)\Microsoft Visual Studio 14.0\VC\include
C:\Program Files (x86)\Microsoft Visual Studio 14.0\VC\atlmfc\include
C:\Program Files (x86)\Windows Kits\10\Include\10.0.10240.0\ucrt
C:\Program Files (x86)\Windows Kits\8.1\Include\um
C:\Program Files (x86)\Windows Kits\8.1\Include\shared

Inherited values:

$(VC_IncludePath)
$(WindowsSDK_IncludePath)

☑ Inherit from parent or project defaults

Macros>>

OK   Cancel
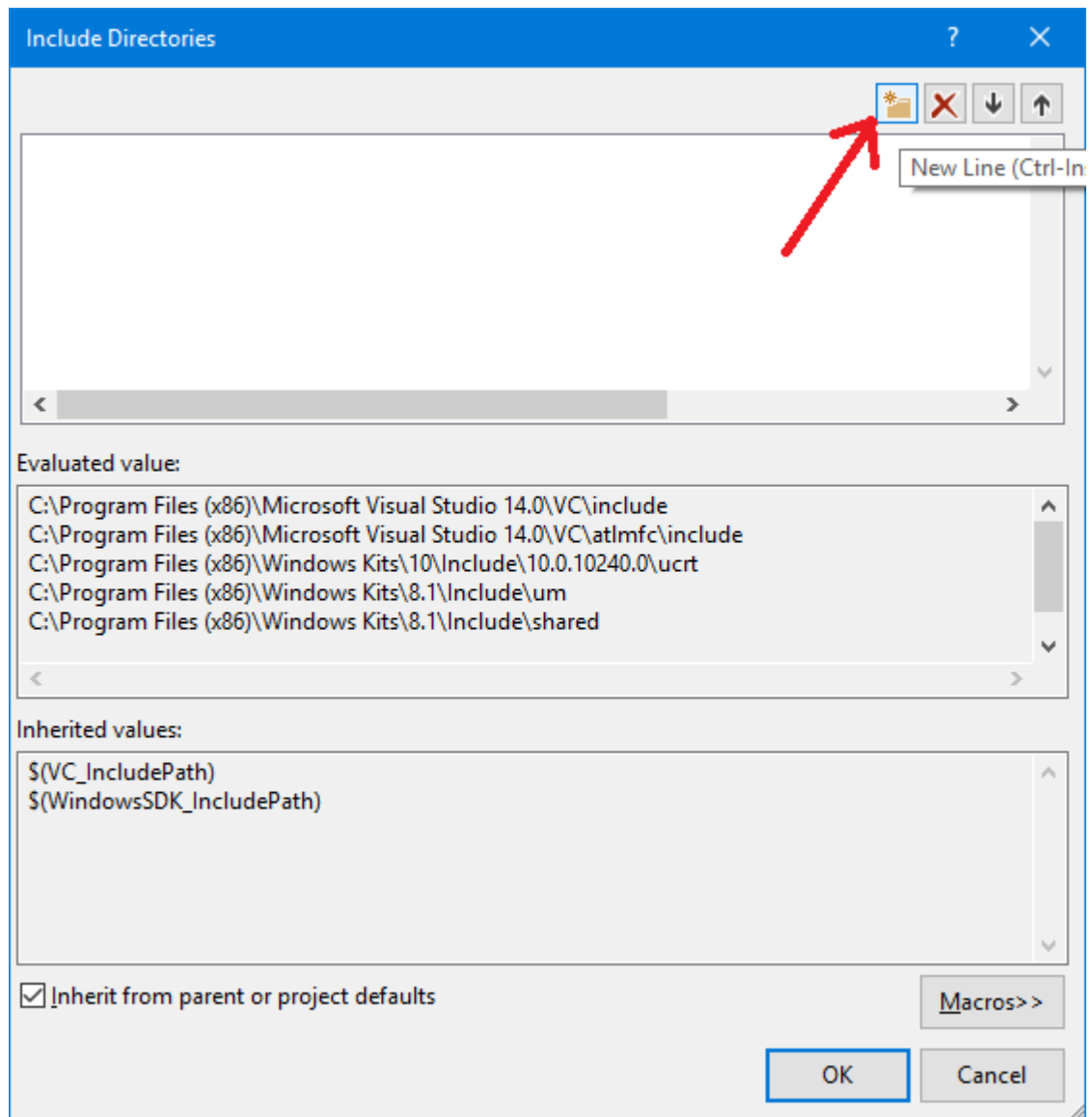
10. Add the path towards the "include" folder in Qt. In my case, this path is:
    c:\Qt\Qt5.6.0\5.6\msvc2015_64\include.

Include Directories

c:\Qt\Qt5.6.0\5.6\msvc2015_64\include\

Evaluated value:

c:\Qt\Qt5.6.0\5.6\msvc2015_64\include\
C:\Program Files (x86)\Microsoft Visual Studio 14.0\VC\include
C:\Program Files (x86)\Microsoft Visual Studio 14.0\VC\atlmfc\include
C:\Program Files (x86)\Windows Kits\10\Include\10.0.10240.0\ucrt
C:\Program Files (x86)\Windows Kits\8.1\Include\um

Inherited values:

$(VC_IncludePath)
$(WindowsSDK_IncludePath)

☑ Inherit from parent or project defaults

Macros>>

OK   Cancel
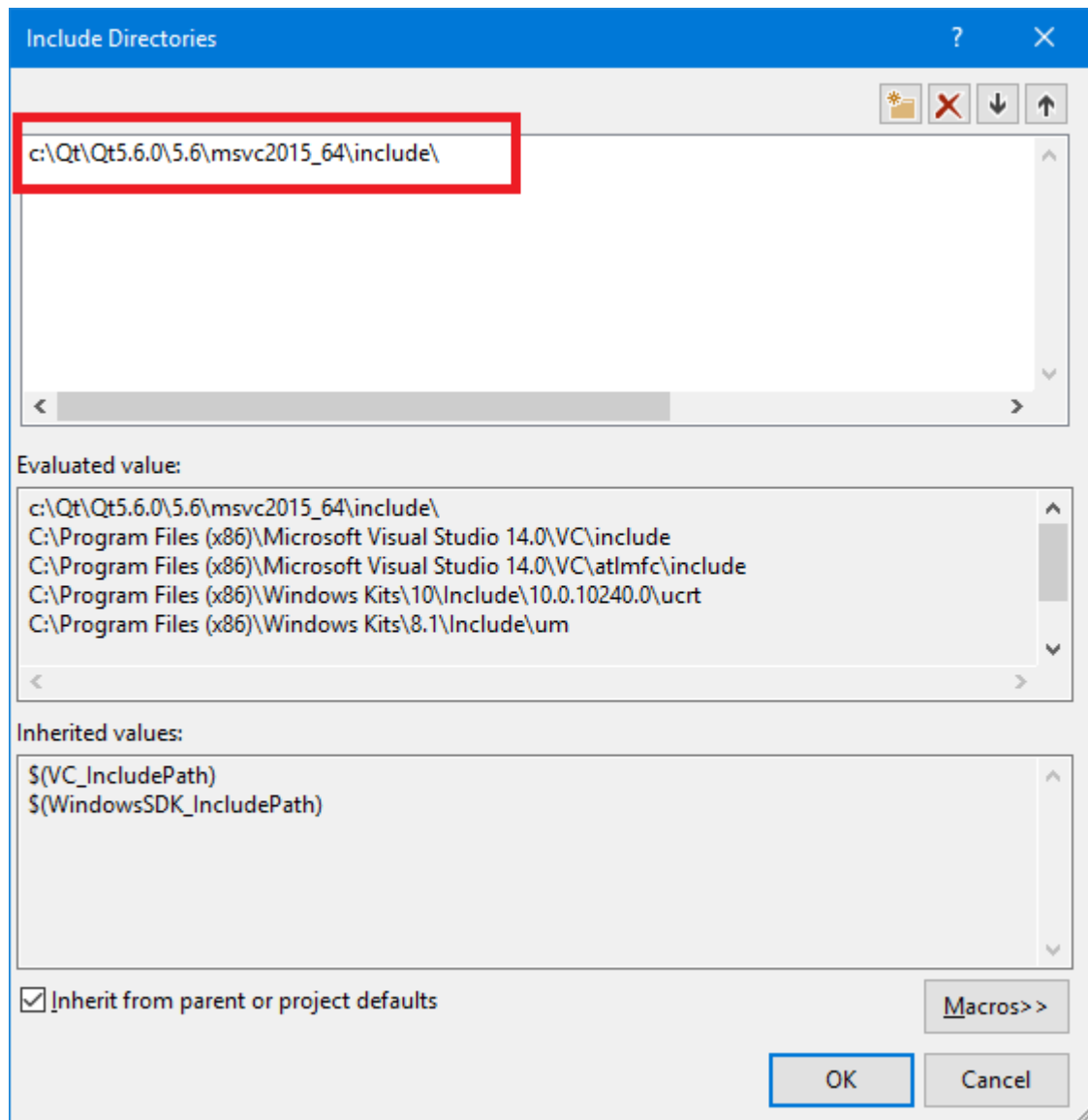
11. The fake "compilation errors" should now disappear. If this does not happen immediately, then restart Visual Studio.

12. Now you are ready to build wonderful GUIs! ☺