**Getting started with C++ in the Introduction To Self Driving Cars Nanodegree with Windows & Visual Studio**

Important note: I do not work at or for Udacity, I am a student in a ND myself and this guide shall just help other students getting an easier start. Typos and errors in this guide are for free, let me know if you found one. :)

EDIT: A friend of mine pointed out that some Americans need to be reminded to not drink boiling coffee and dry cats in the microwave and might call Saul Goodman afterwards, so here it comes ;-):

Disclaimer:

THIS GUIDE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

You may freely share the link to this document.

**© 2017 by Michael Ikemann, Head of Mobile Engineering @  
**

[**www.logiball.de**](http://www.logiball.de)

[@Alyxion](https://twitter.com/Alyxion)

[Michael Ikemann @ LinkedIn](http://www.linkedin.com/in/michael-ikemann)

[Click here for the Mac guide](https://docs.google.com/document/d/1FoIc9X0NcM-0y0dgXCzJcpu5lVXGFqS62mb1ABLDT1M/edit?usp=sharing)

[**1. Installing Visual Studio**](#_vky4vrudamhc) **2**

[**2. Recommended tools**](#_d7v3osangowa) **2**

[**3. Download the project files from the Udacity class room and extract them**](#_mu3ec9z11bma) **3**

[**4. Start Visual Studio**](#_3pvmm9blf7a2) **3**

[**5. Project setup**](#_k01a8znjg0hx) **3**

[**6. Back to Visual Studio**](#_81w4obk7j62s) **6**

[**7. Precompiled headers**](#_irm1va9ptp8w) **10**

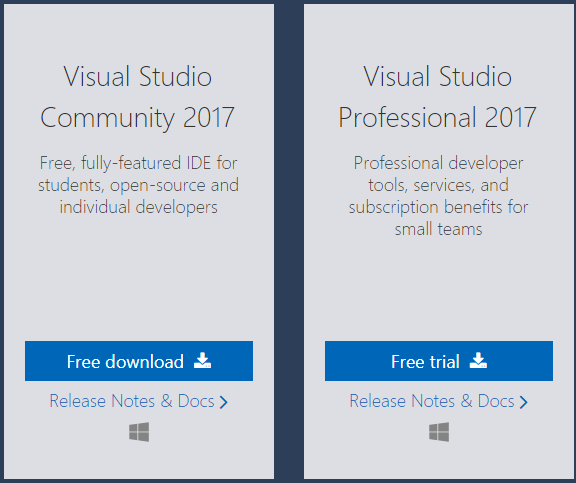
[**8. Finally…**](#_5geuy2wo3t4a) **10**

[**9. Debugging**](#_kpcsx7ymp0wt) **11**

### 1. Installing Visual Studio

First of all you of course need to install Visual Studio 2017. Note that there is also a version called Visual Studio Code which has basically nothing in common with the “real” Visual Studio except the name. The “real” one is available for Windows only. While the installation process make sure you install all common C++ features.

<https://www.visualstudio.com/downloads/>



### 2. Recommended tools

If you want to professionally develop software, you should (at least in the long term) consider the two tools Visual Assist X and Total Commander.

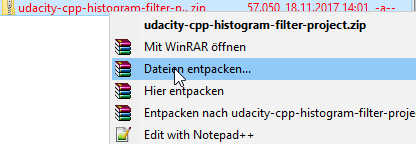
Total Commander is a very nice to have filemanager made for geeks and/or s/w developers, because all of it’s file handling, comparison and sync features. You can (if you have installed TortoiseGit) also let it show all file states and extend it with built-in git features like showing the branch of all projects and so on and so on. :)  
Visual Assist X will dramatically accelerate your software development. Sounds like I am a vaccum salesman, but learn all it’s hotkeys, refactoring, code insight and automatism features and you will start weeping silently whenever you have to develop for OS X, iOS or Linux, where Visual Studio and so Visual Assist X are not available :P.

<https://www.wholetomato.com/downloads/>

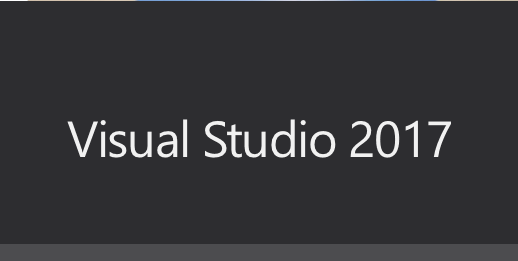
<https://www.ghisler.com/index.htm>

You can use Total Commander for free forever if needed and the 30 day trial phase of Visual Assist X should likely be enough for evaluating it.

### 3. Download the project files from the Udacity class room and extract them

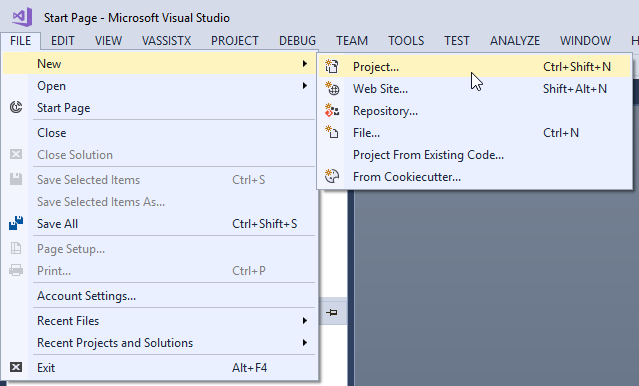


### 4. Start Visual Studio

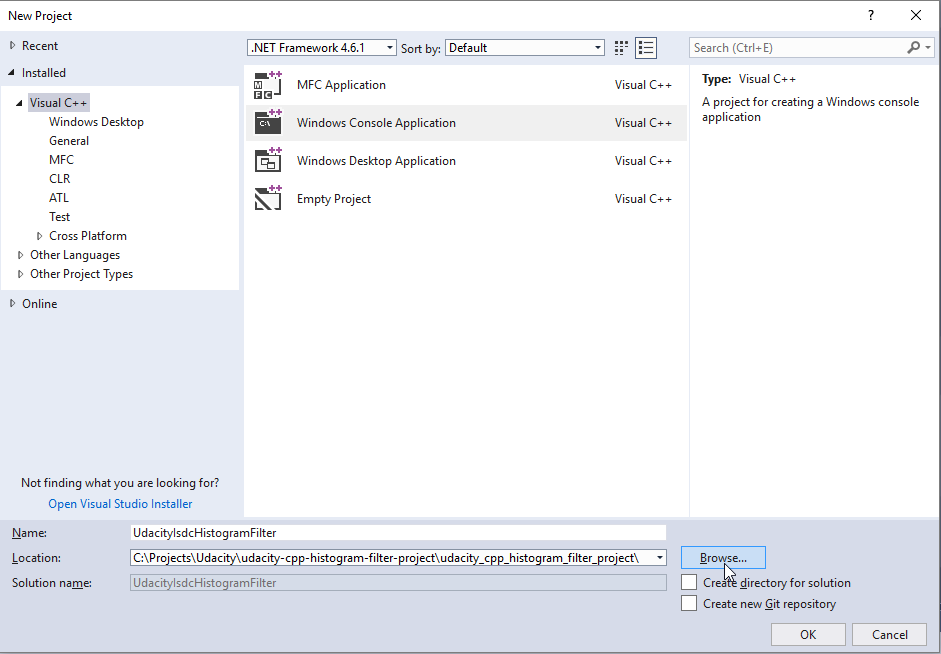


### 5. Project setup

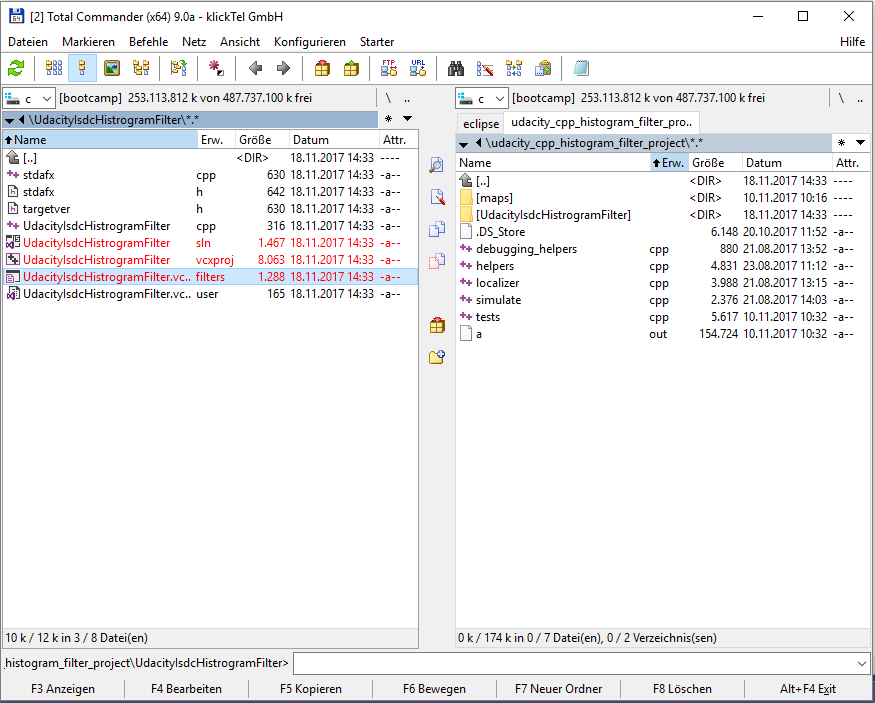
* Click on File → New → Project



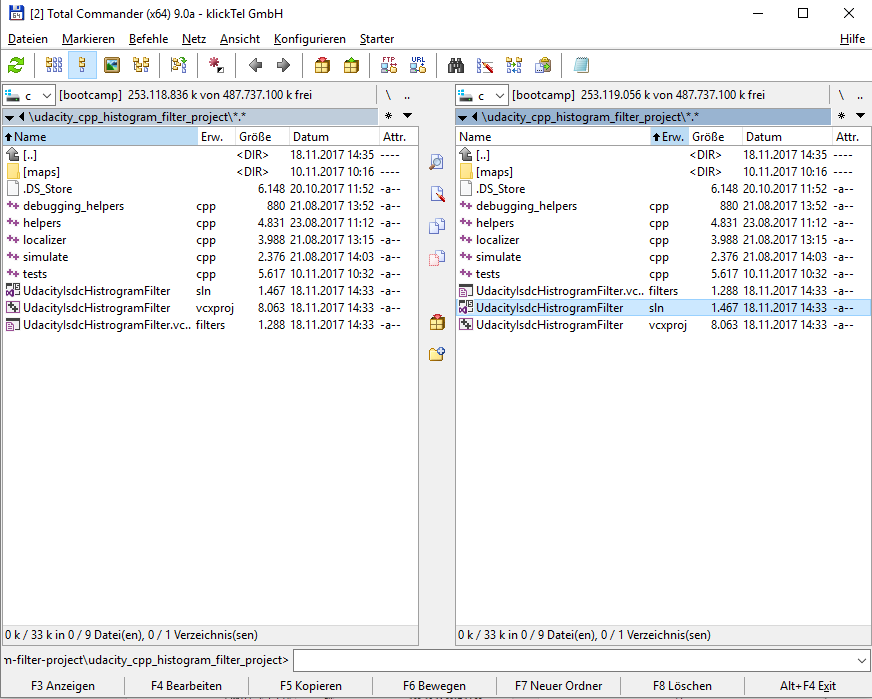
* Select **Visual C++** on the left side (not any of its sub features!) and **Windows Console Application** on the right side
* Enter a name for the project
* Click **Browse** and select the folder into which the files tests.cpp etc. have been extracted, so the lowest level.
* Press OK



* Close Visual Studio now completely
* Start Total Commander (if you installed it), move to the new folder containing the sln, vcxproject etc. on the left and to the folder containing Udacity’s cpp files on the right side
* On the left side you can now select the .sln, the .vcxproj and the .filters files using the space key, they should get highlighted like shown below. Now press Move (F6) to move them to the right side.

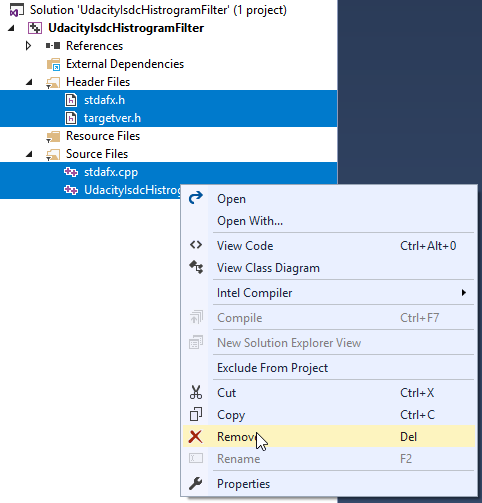


If everything went fine the sln, vcxproj, filters and all cpp files should be within the same directory now. You can delete the subfolder UdacityIsdcHistogramFilter.

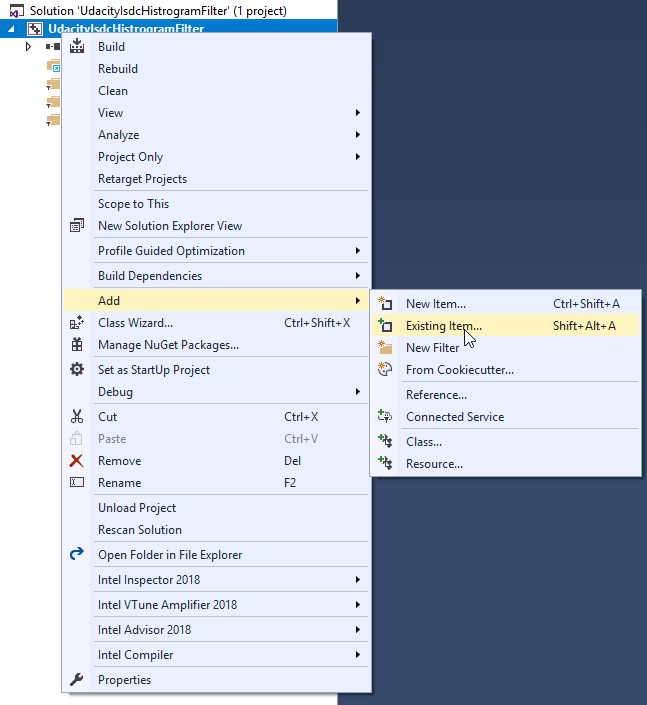


### 6. Back to Visual Studio

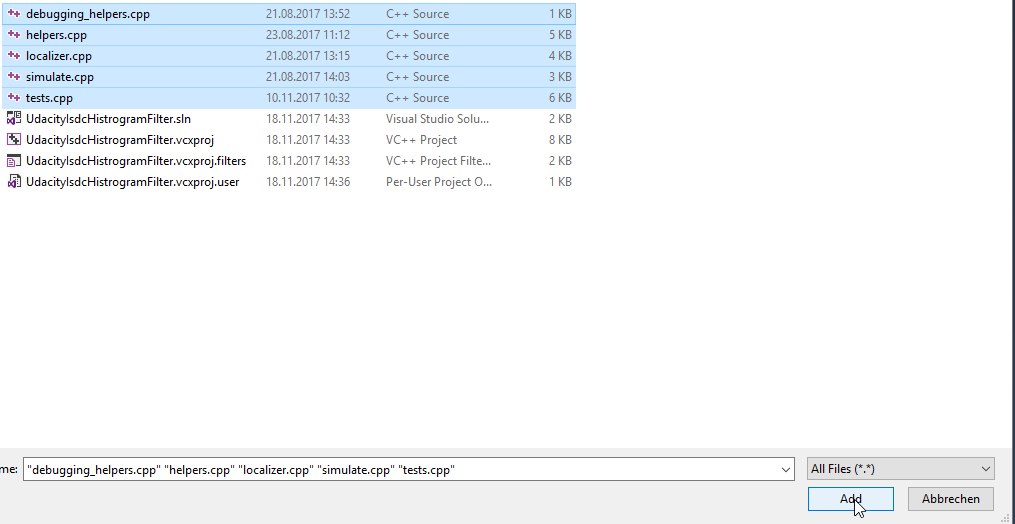
* Double click on the sln-File to start Visual Studio again.
* Because the project fille doesn’t know anything about our “sabotage” yet, we have to select the 4 files (just the 4 files, not the folders, use Ctrl key), right click and remove them.



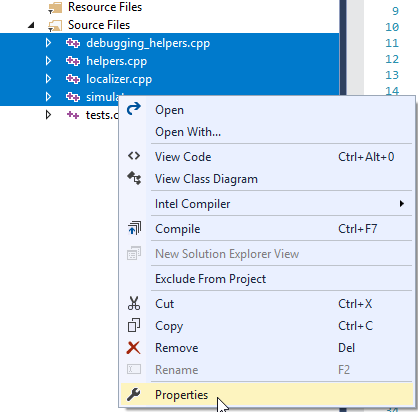
* Now right click on the project name on the left (written in bold) → Add → Existing Item



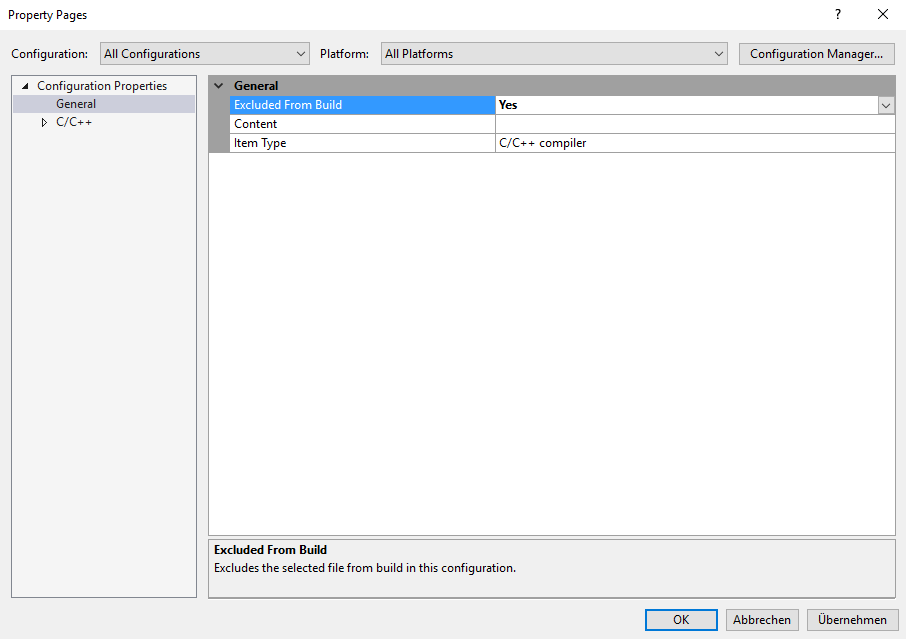
* Because the project and our cpp files share the same folder you should see them directly. Select all 5 cpp files and press the “Add” button.



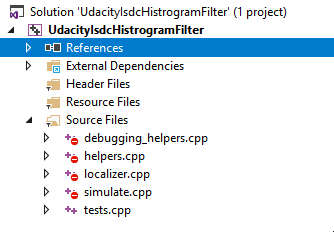
* Because all of them are cpp files they should now get listed in the SourceFiles filter on the left side.
* Select all files **except tests.cpp** like shown below. Because this is a single-file-linkage-project we are only allowed to compile tests.cpp which includes all other files by itself.



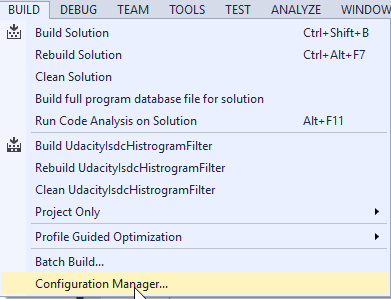
* Right click into the selection and then on **Properties**
* Select **All Configurations** for Configuration and **All Platforms** as Platform at the top
* Select **General** on the left side
* Switch **Exclude From Build** on the right side to **Yes**
* Press **OK**

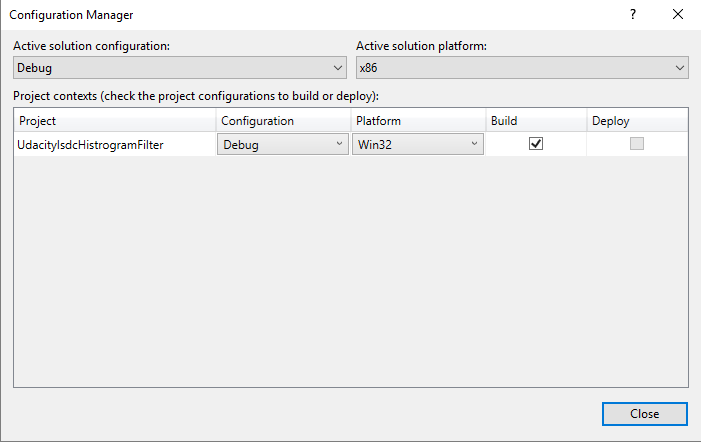


* Afterwards the 4 files you previously selected should now be shown with a red sign next to them, signaling that they are ignored by the compiler, NOT by the CodeInsight though and this way you can still reach them fast to edit them.



* I experienced that Visual Studio likes to change the initial configuration sometimes, majorly if you had a previous installation of VS one day. Just make sure now via Build → Configuration Manager that the **active configuration** is set to **Debug** and the **active platform** to **x86**.

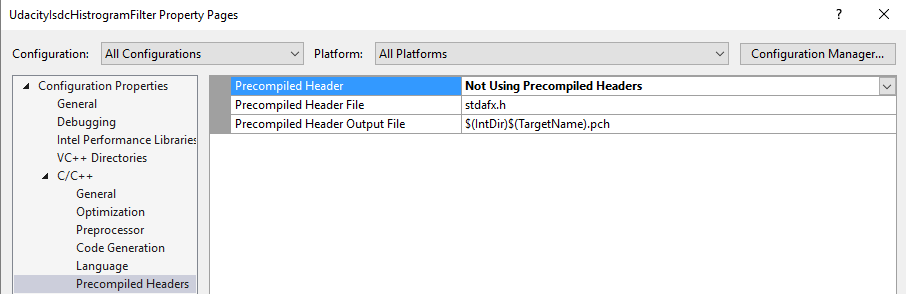




### 7. Precompiled headers

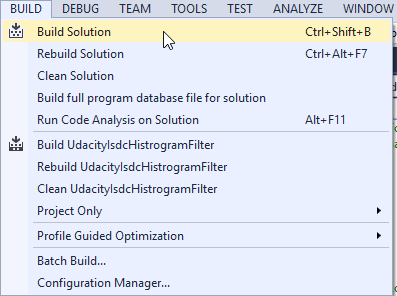
By default Visual Studio setups all it’s projects for using precompiled headers. This also makes a lot of sense if projects become larger and require more time for compilation. But to keep it simple and the code compatible to other platforms, we have to deactivate them now.

* **Right click on the project’s name** (written in bold) on the left side and then **Properties** again
* Select **All Configurations** and **All Platforms** at the top to make sure it will also work fine in release mode and for x64.
* Select now **C++ → Precompiled Headers** on the left
* Toggle **Precompiled Header** to **“Not Using Precompiled Headers”** as shown below
* Press **OK**

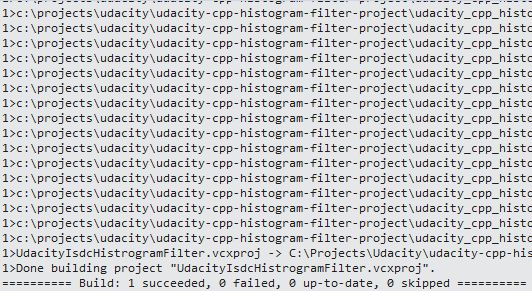


### 8. Finally…

We are nearly done. Click on **Build** → **Build Solution**

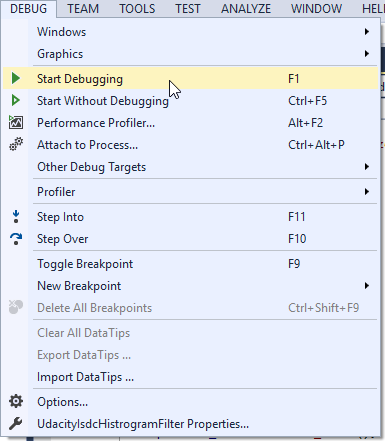


You should get a lot of warnings (unfortunately) and at the end a line hopefully showing now “Build: 1 succeeded”.



### 9. Debugging

* **Click on Debug → Start Debugging** now



You will see a window for half a second. That’s great, because the app seems to run and bad, because you don’t see what it says. You can click on “Start Without Debugging” which starts the app via a terminal and lets you press a key after its execution, but therefor are not able to debug anymore.

One of the many workarounds for this is the following:

Open tests.cpp and insert these 5 lines at the end of it’s main function at the bottom.

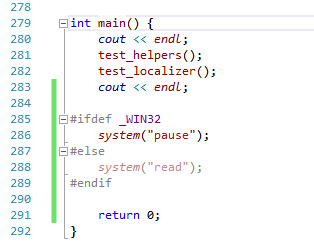
#ifdef \_WIN32

system("pause");

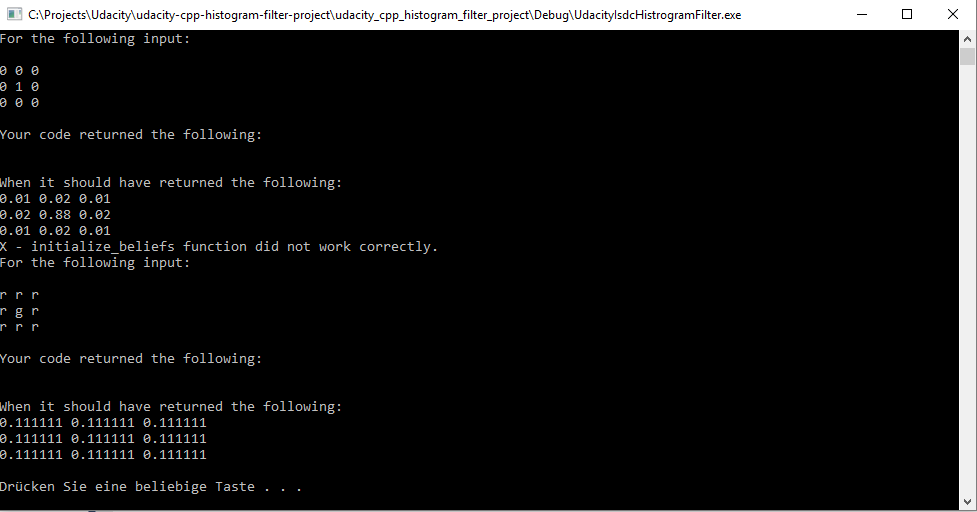
#else

system("read");

#endif



And taaadaaa… the application also waits in debug mode for your confirmation.



**I hope this guide helped you to get a fast start with the C++ development in Visual Studio.**

**Have fun and happy coding!**