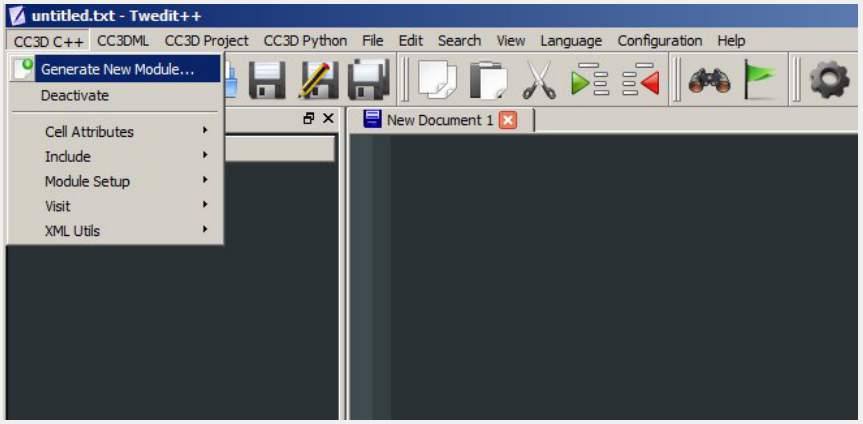
Automating CC3D Module Development

Once you learned how to compile CC3D code from source you probably would like to develop some C++ CC3D plugins. The good news is that this task can be greatly streamlined with the help of Twedit++. In this example I will show you how to generate template code for CC3D plugin (generating steppables works the same way).

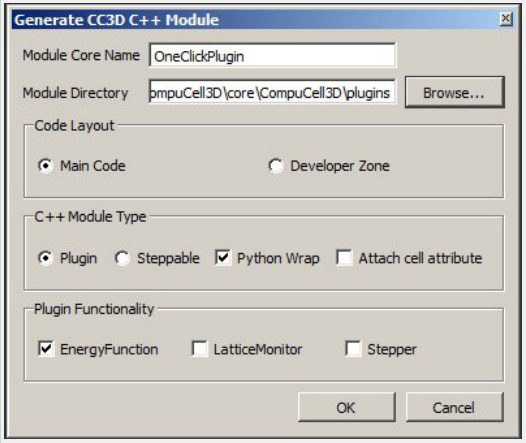
Before we start I recommend that you make a copy of the CC3D source tree. The reason for that is simple, in case things go wrong you do not want to end up with messed up source directory.

The source directory that I am will work from is in **D:\CC3D_GIT_DEMO** and as you can tell I am on Windows machine. However all the things presented here apply to any platform so please do not panic if you use OSX. I mean, obviously this is something to reflect on later but for now we assume that OSX is acceptable.

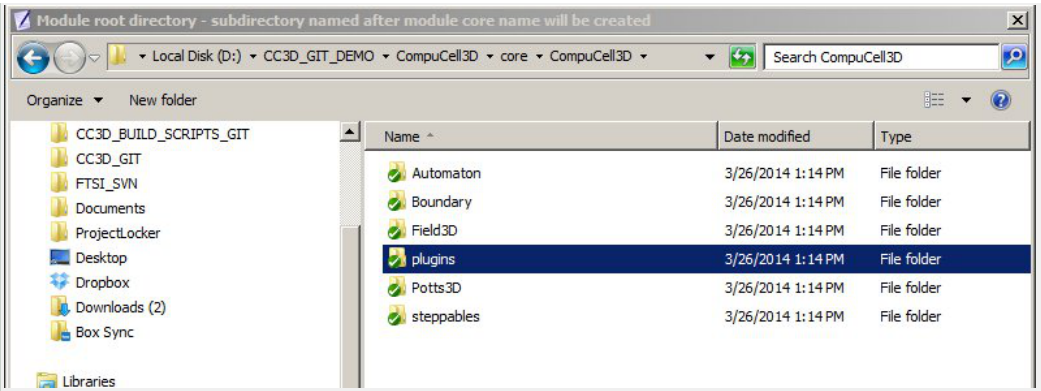
Let us begin. We first open Twedit++ and go to **CC3D C++** menu and choose **Generate New Module..**:



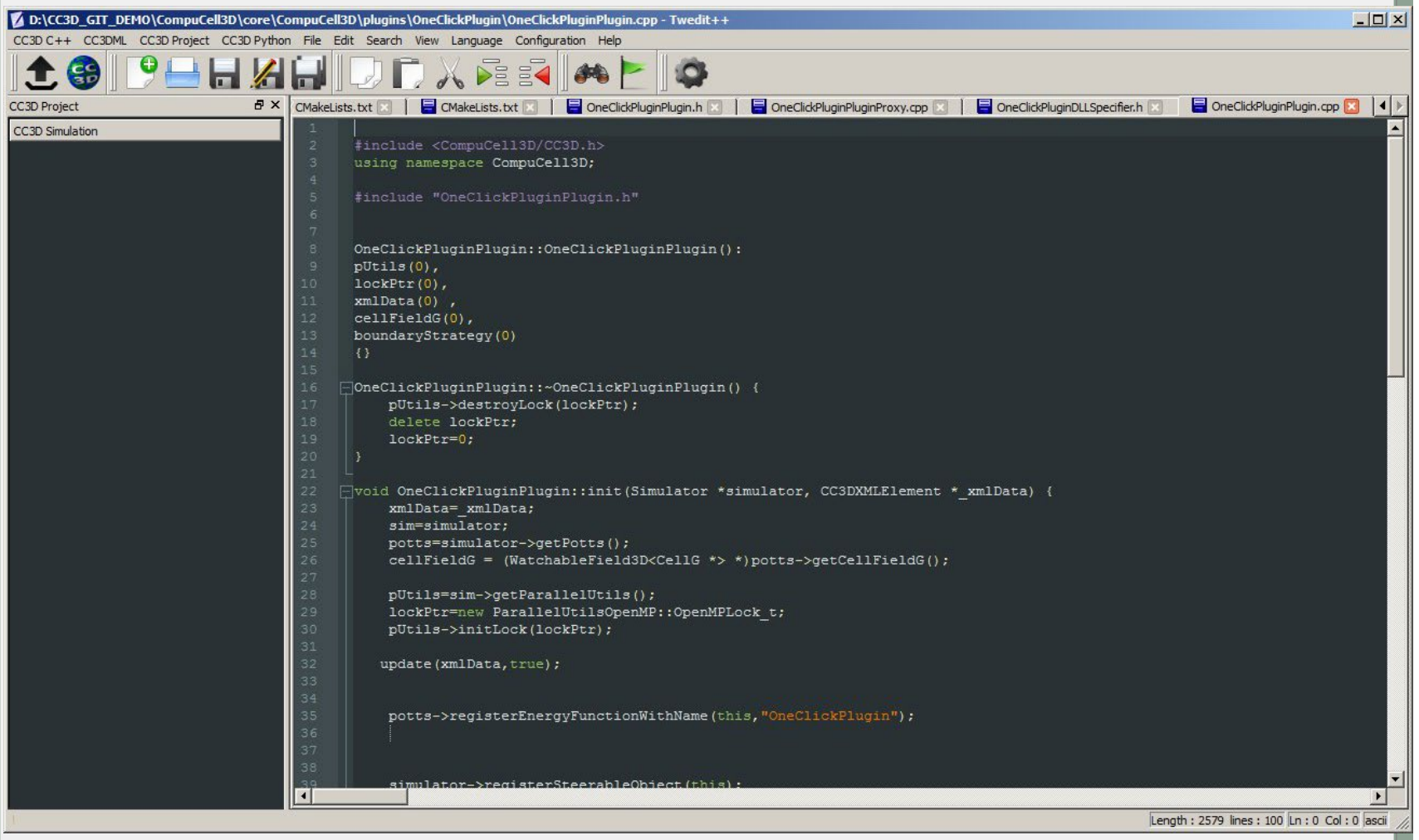
A dialog will pop up that has several options on it. We have to fill the plugin name and point to the folder where plugins are stored



As you can see I named the plugin **OneClickPlugin** and set Module Directory to point to **d:\CC3D_GIT_DEMO\CompuCell3D\core\CompuCell3D\plugins**:

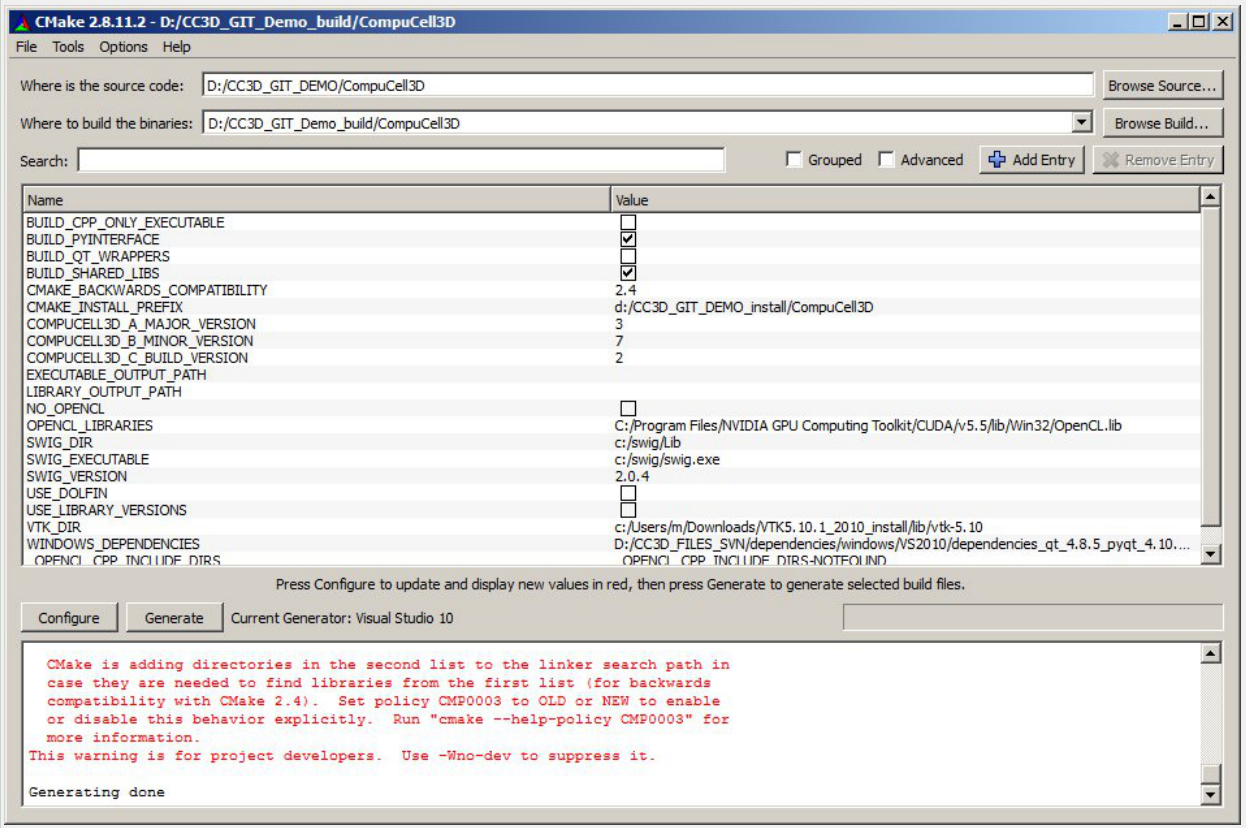


In the Generate New Module dialog (see above) I have checked **Main Code** to indicate that the plugin will be stored in the main source tree (alternatively you can generate new plugin in the Developer Zone folder). i have also indicated the module I want to generate is a CC3D plugin and that I want it to be Python wrapped - meaning to be accessible from Python. Python Wrapping is optional and if you are sure you will use new plugin by specifying its configuration in CC3D XML than you may uncheck **Python Wrap** option. I also specify that the plugin will be an energy function type. By checking Stepper and Lattice Monitor you can add additional functionality to the plugin - see CC3D Developers Guide for more details . These are all the things we have to do to enable code generation. Click OK and plugin files get generated:

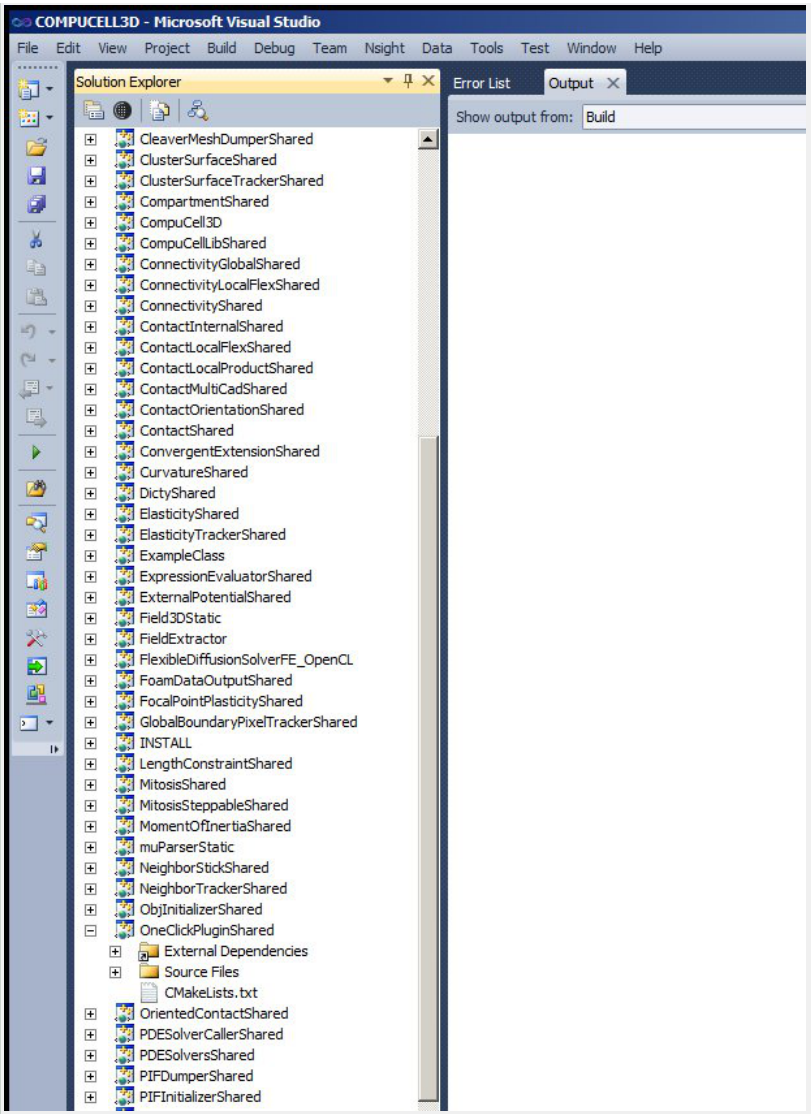


In this particular example Twedit++ generated (or modified) 10 files. As you can see from the attached screenshots files that were modified were CMake configuration files and SWIG configuration files. Files that were generated included OneClickPlugin source files and of course its CMake configuration file.

because we generated/modified CMake configuration file we need to rerun Cmake and generate our project again:



After we open the project in Visual Studio (or Kedevelop or Eclipse) we can see that **OneClickPlugin** appeared as a CC3D module.



Now we can build CC3D and modify **OneClickPlugin** to make sure that it does something useful. For details how to develop CC3D plugins please consult CC3D Developers Guide

= Generating Steppable Template Module==

To generate Steppable template code we follow essentially the same steps as above but we choose Steppable in the **Generate New Module *dialog, name if* *OneClickSteppable* *and point to steppables directory*** *d:\CC3D_GIT_DEMO\CompuCell3D\core\CompuCell3D\steppables:*