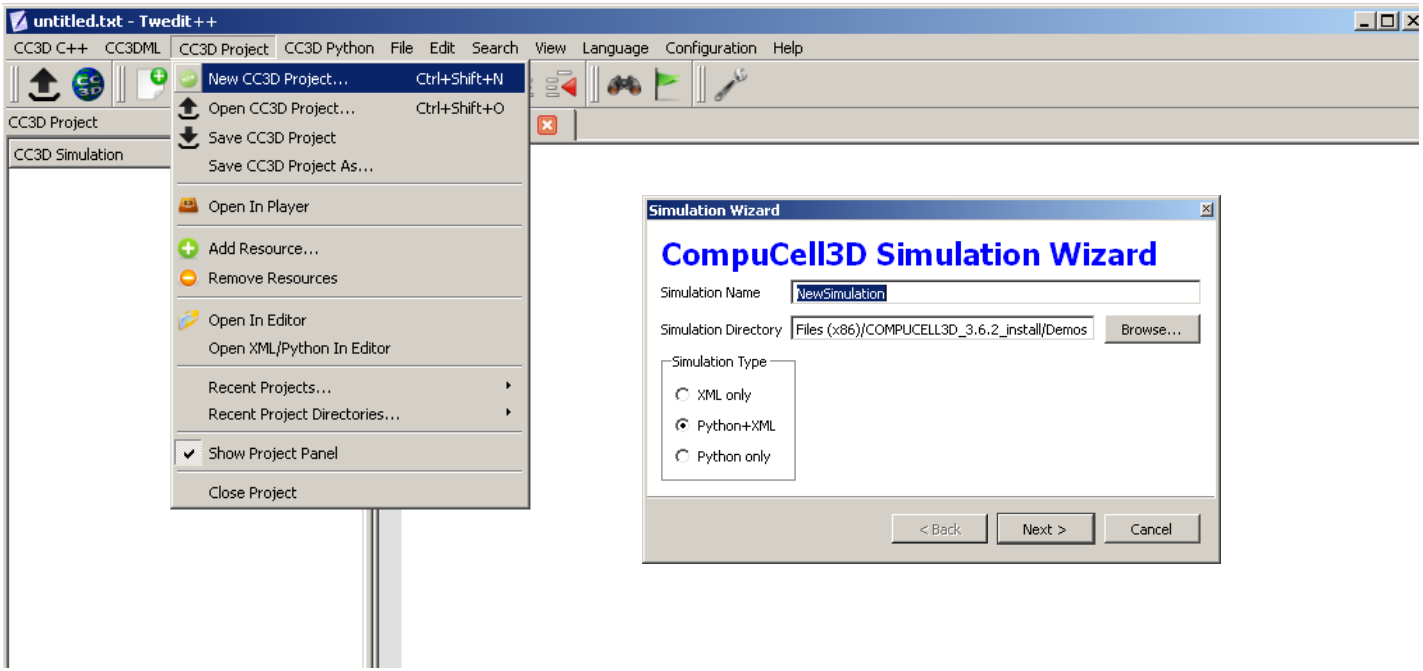
# Introduction to Twedit++-CC3D

Maciej Swat

# Why use Twedit++-CC3D

- CompuCell3D customized version of twedit++ programmer's editor
- Simplifies development of simulations by providing Simulation wizard, Python and CC3DML code assistants
- Speeds-up significantly development of CC3D C++ extension modules (plugins and steppables). It sometimes makes sense to replace slow Python steppable with fast C++ counterpart

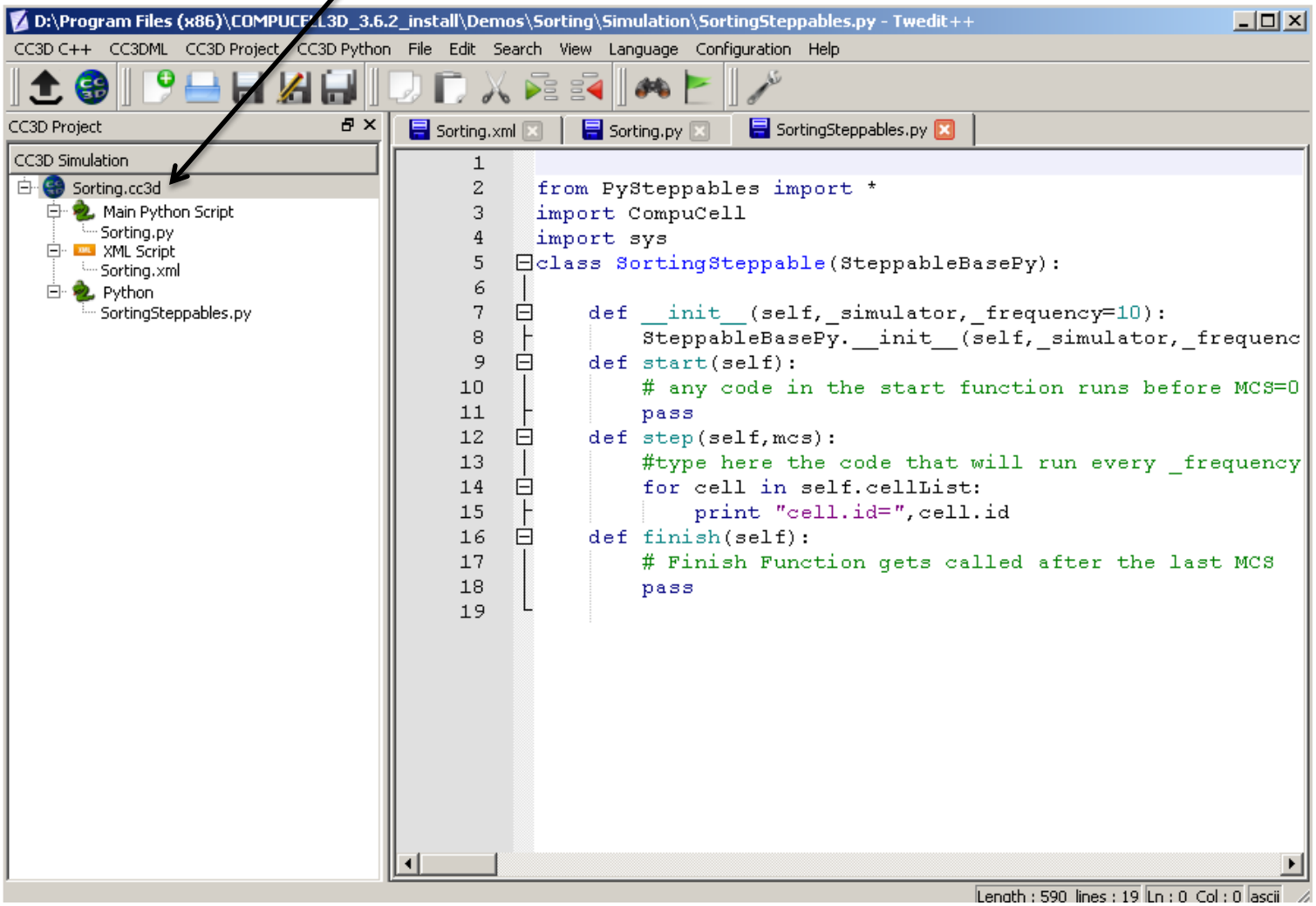
# Starting New CC3D project



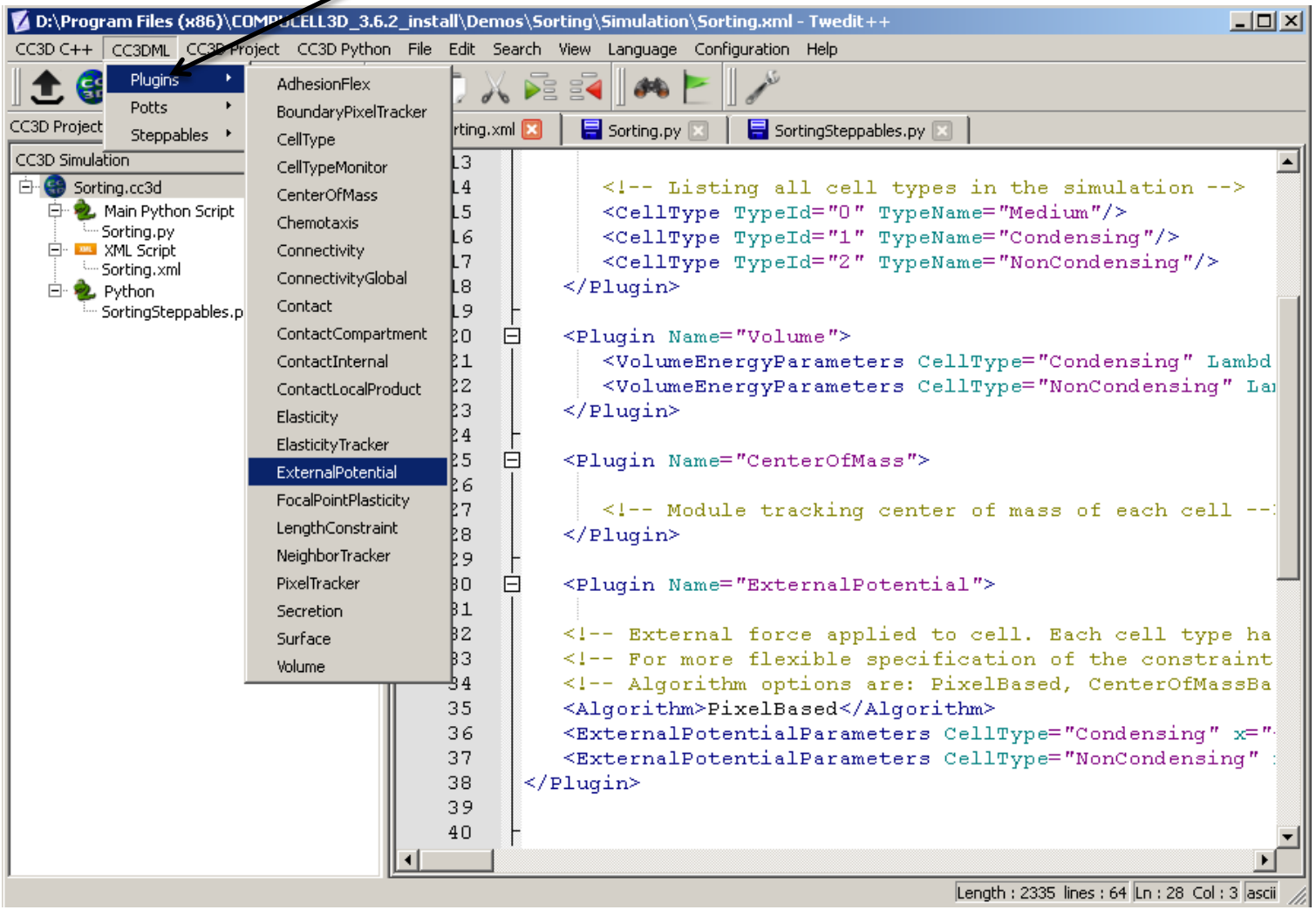
# Simulation Wizard

- In simulation wizard you specify cell types, diffusing fields, cell properties etc…
- If you forget something , do not worry, Twedit++-CC3D code assistants will help you put everything you need into your simulation code

Double-clicking on project name opens all project files in Twedit++



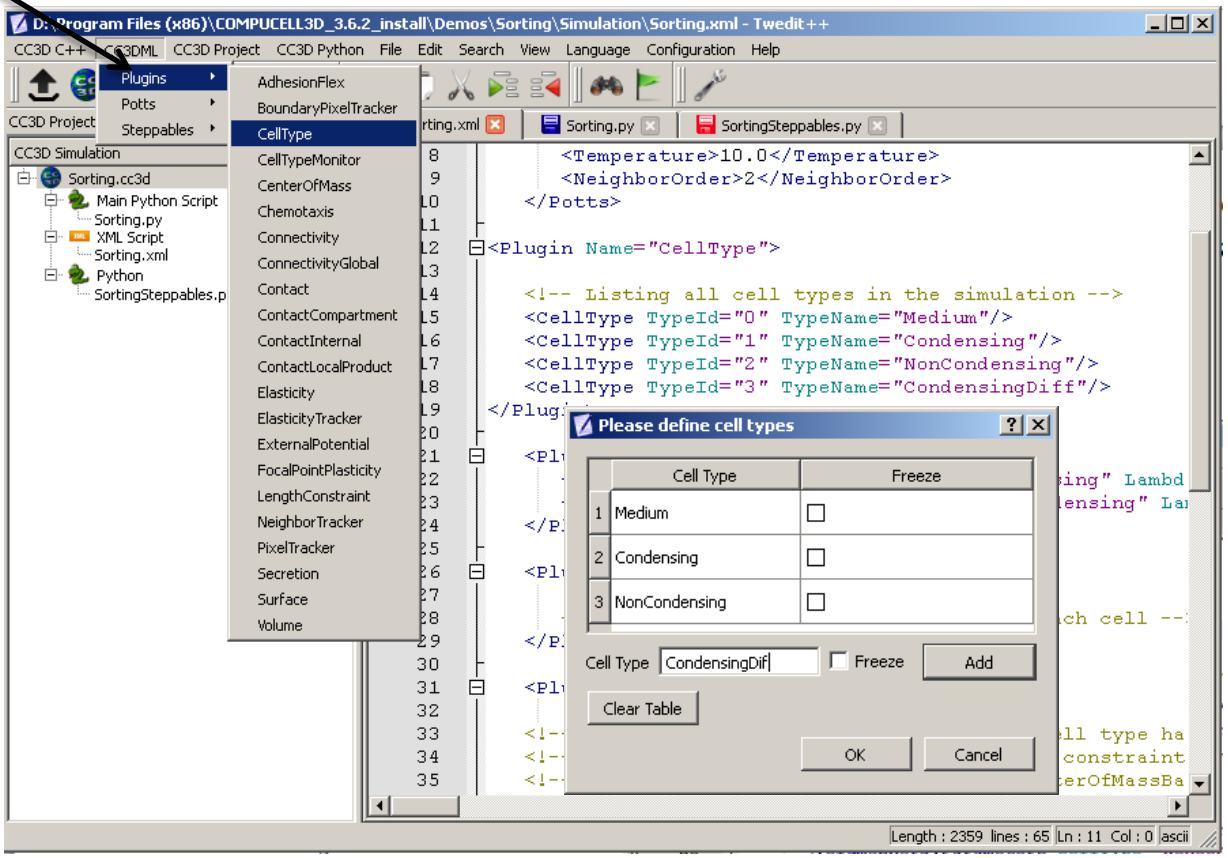
Cell behaviors can be added after completing wizard-based simulation autogeneration. Here we add ExternalPotential plugin which exerts a force on cells. Most CC3D modules are covered. If a plugin is missing from pull down menu it can be added to Twedit pretty easily. Please let us know and we will do it for you.



Adding/removing cell types is most hated operation because it requires modification of many simulation components. Twedit++ makes this operation pretty painless and it does so without using sedatives:

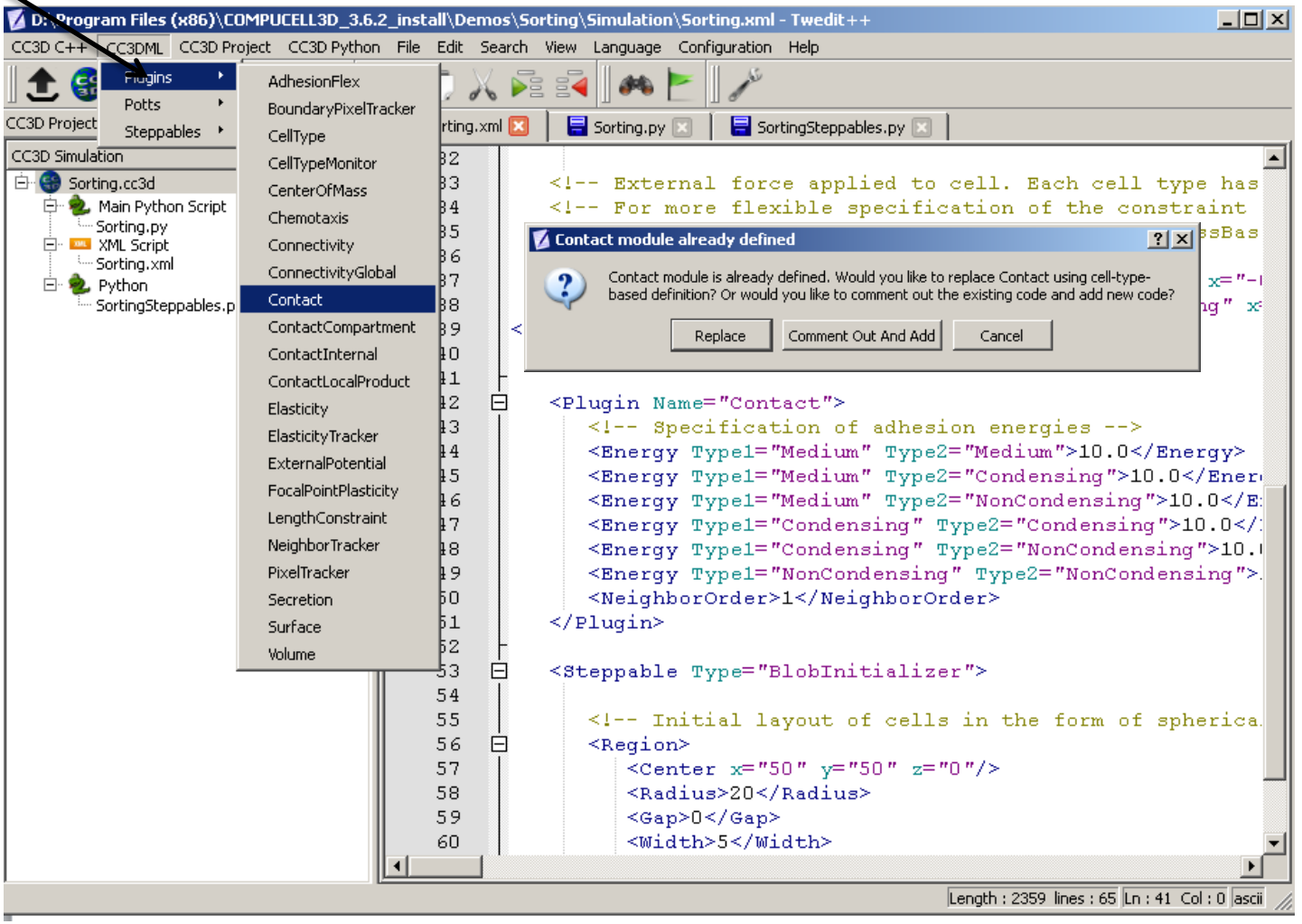
Go to CC3DML->Plugins->CellType and insert new type.

**Important:** After adding removing cell type you have to change CC3D module definitions which use cell type names

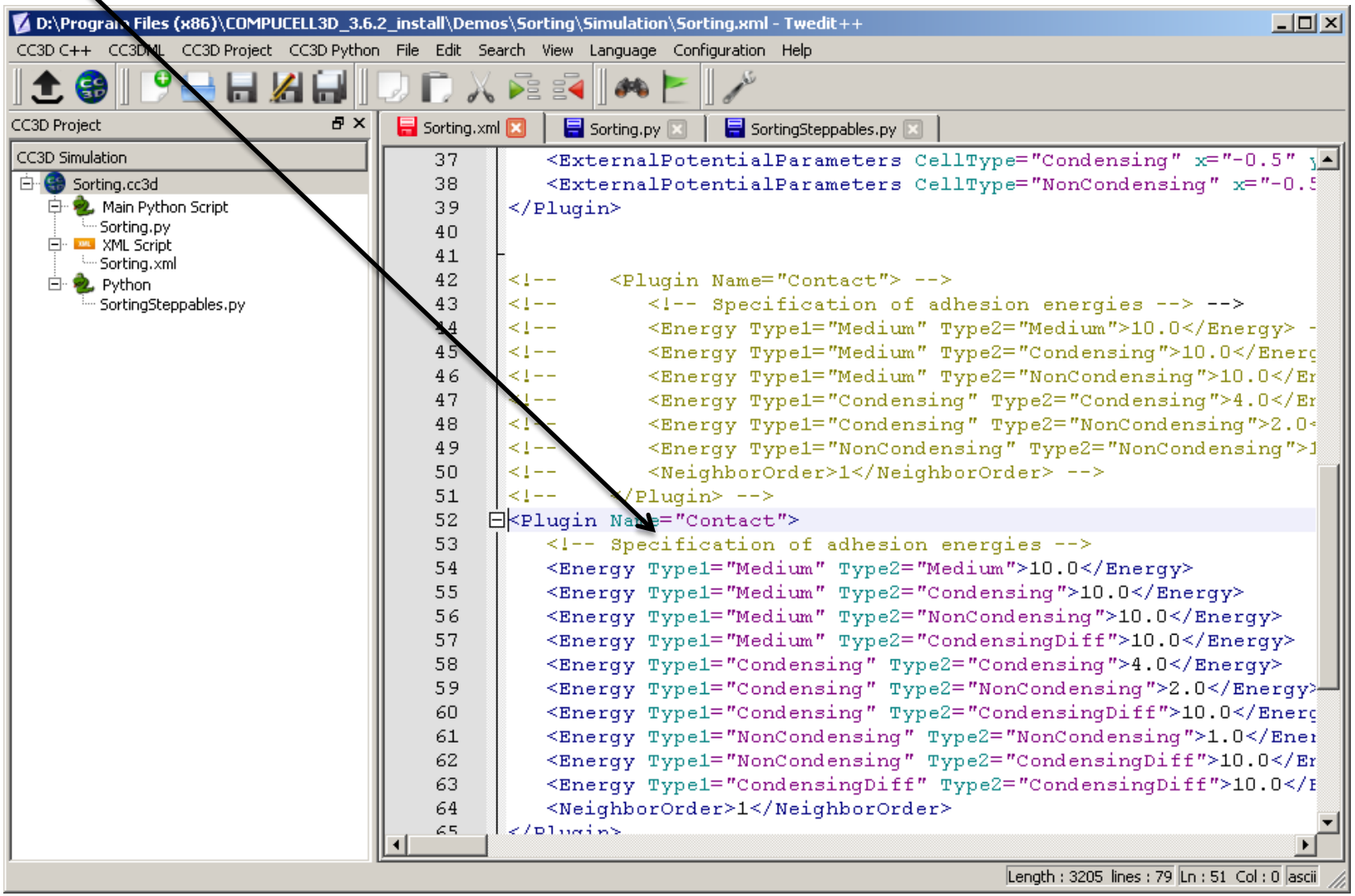


Changing plugin definition which list cell types is relatively easy . Most plugins are supported although occasionally you may need to do some typing:

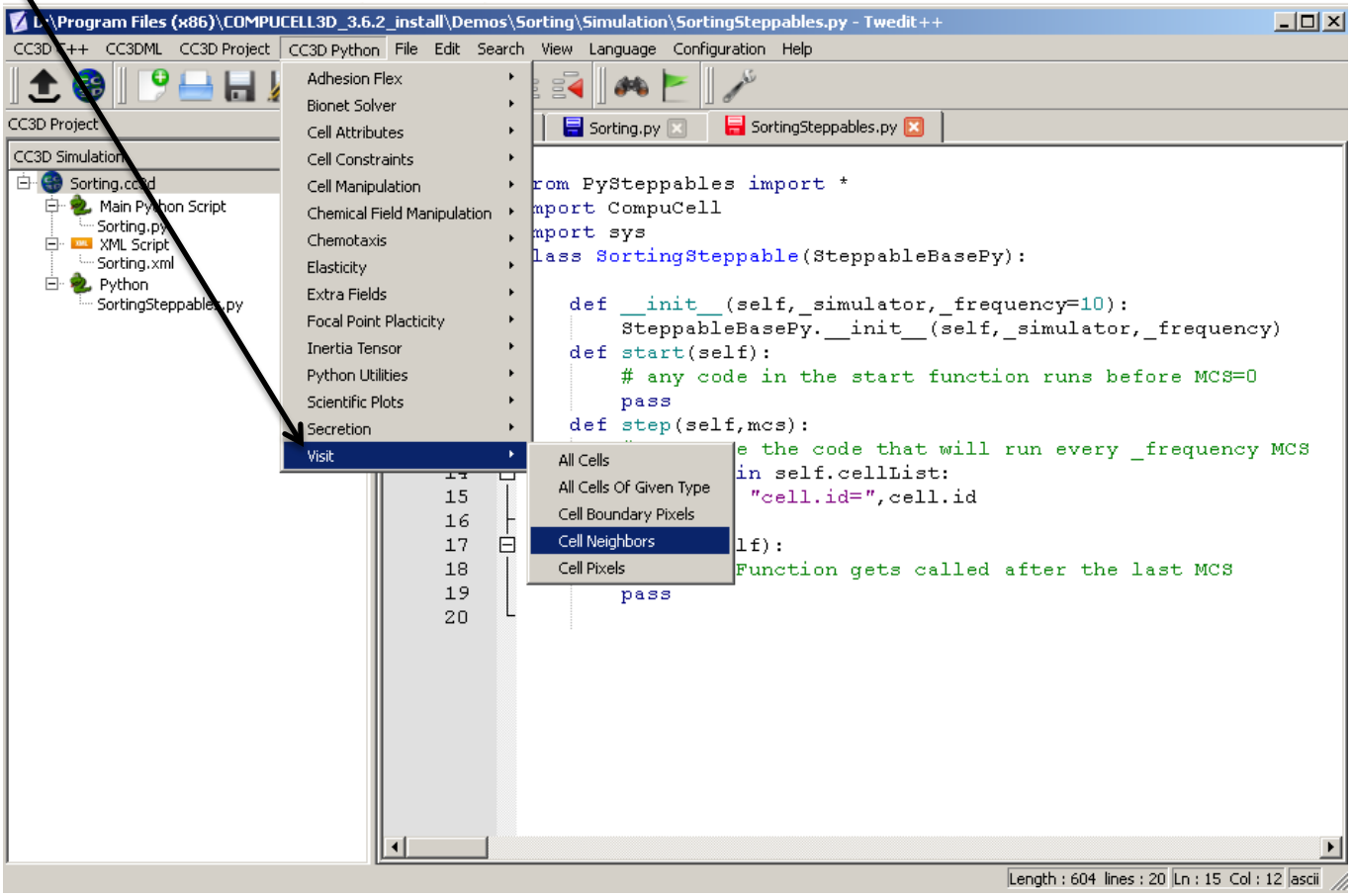
For Contact plugin go to CC3DML->Plugins->Contact and you will get a choice of either replacing or commenting out existing code (if the code for contact is already there)



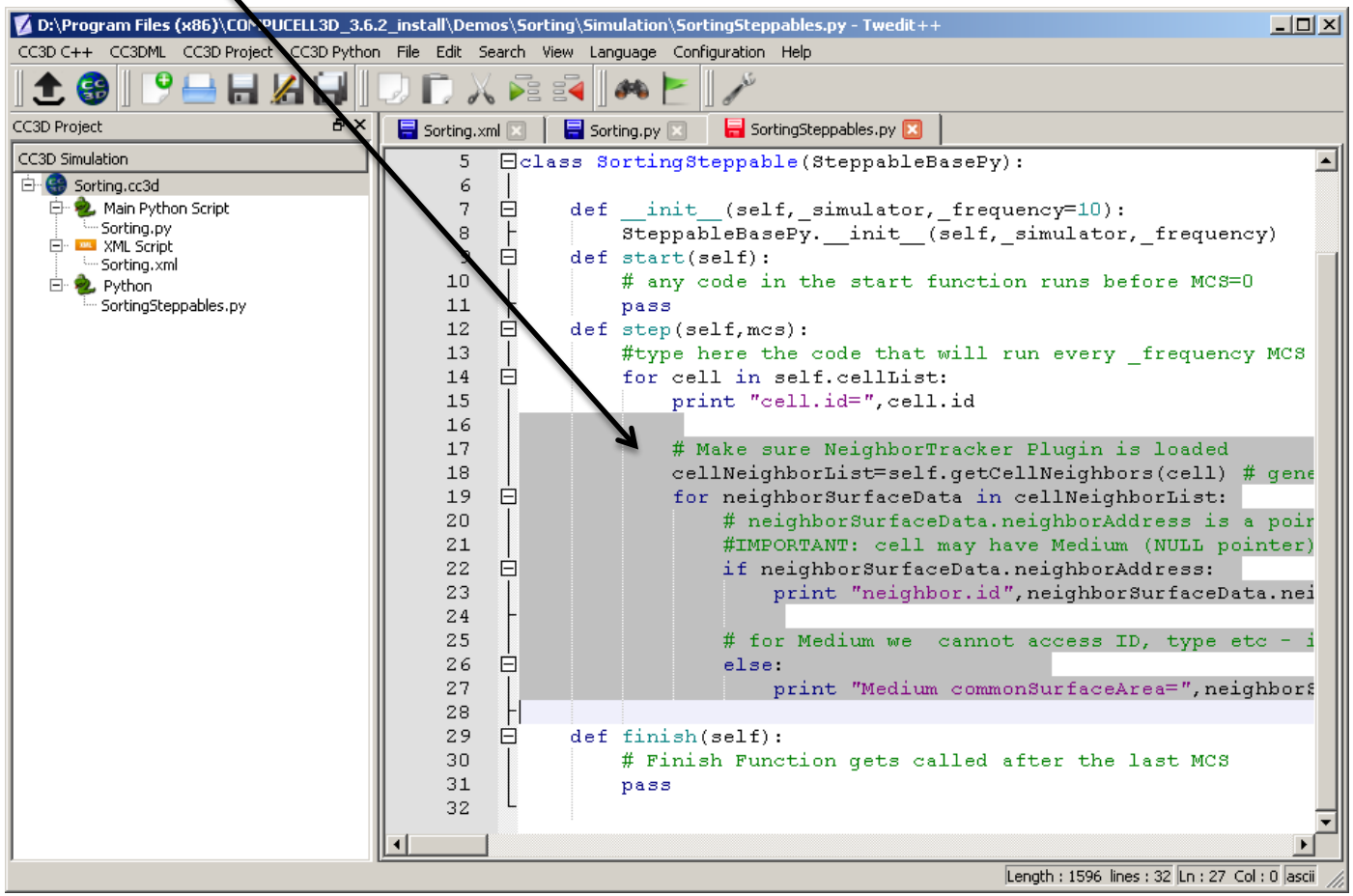
New code for contact includes new types we have defined but contact energy values between types present in the old simulation code are preserved – less typing for you



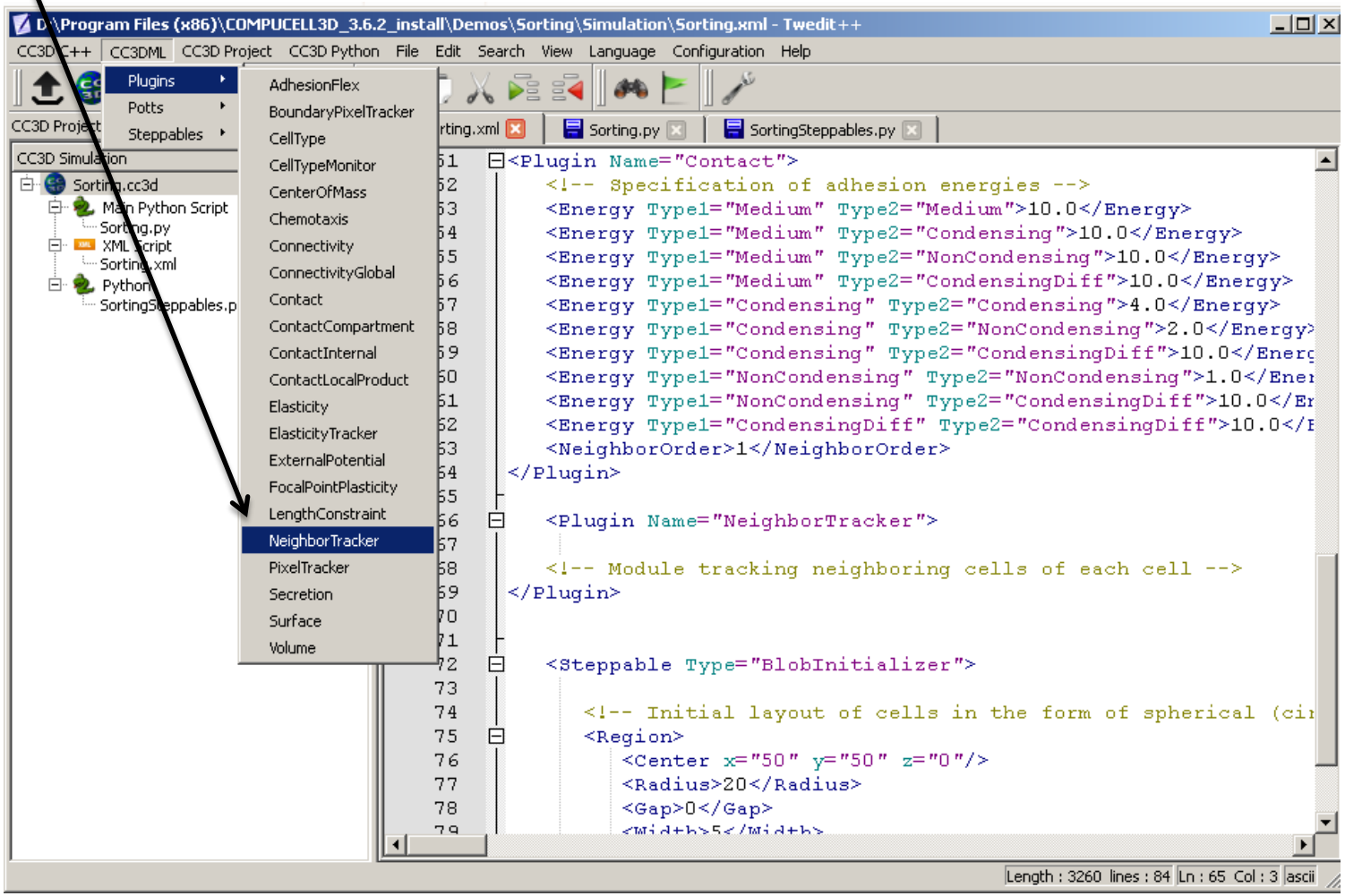
Python CC3D code assistant inserts code snippets for the most common CC3D tasks. Here we add iteration over cell neighbors



Cell neighbor iteration snippet. We usually have to do small edits to adapt it to our simulation



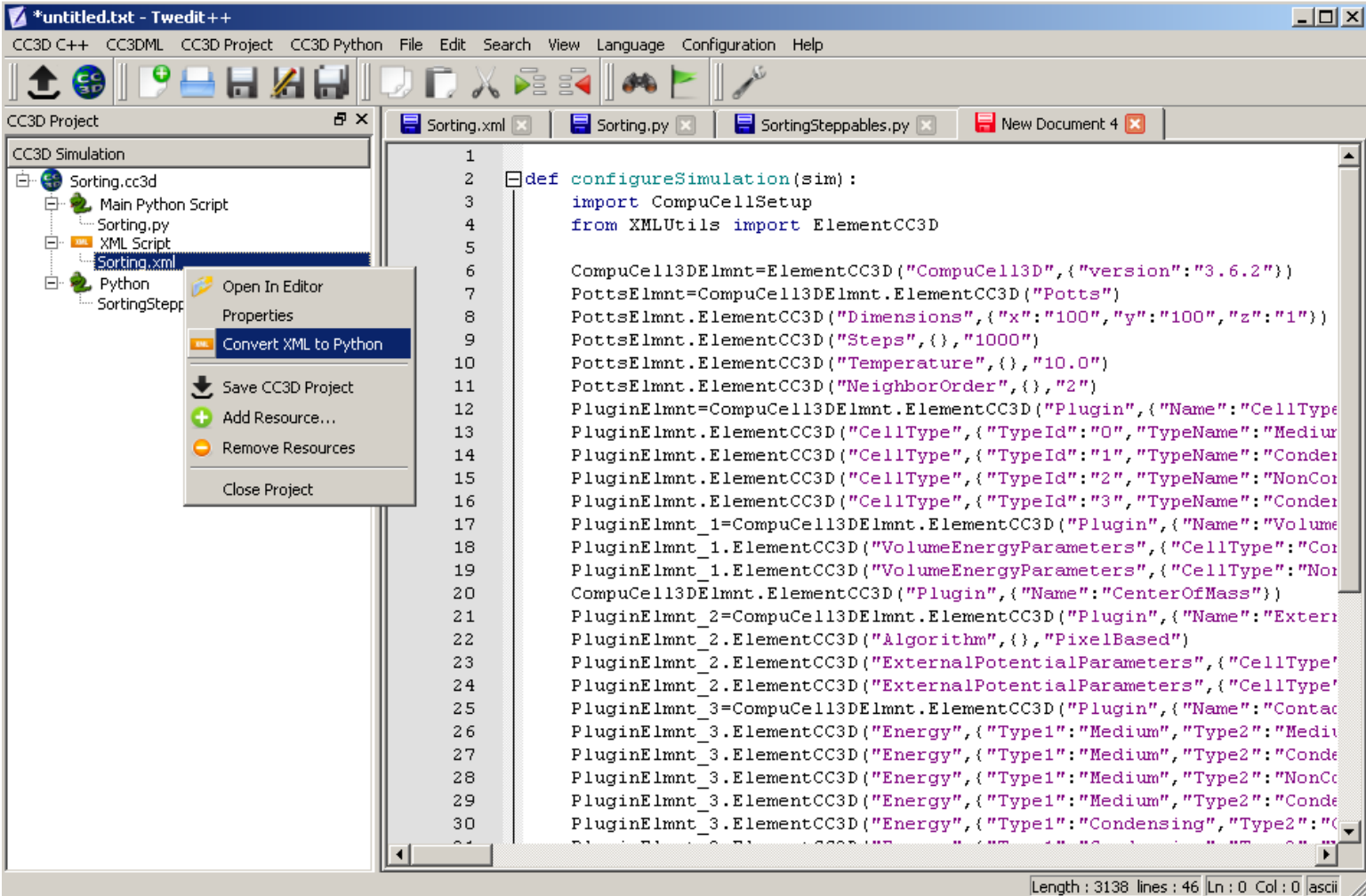
And we also must make sure that we insert NeighborTracker plugin in CC3DML:



# One-click conversion of CC3DML to equivalent Python syntax

Right-click (in the CC3D Simulation panel) on the .xml file you wish to convert and choose ‘Convert XML to Python’. There are two things worth noticing here:

1. One-click solutions in reality require multiple clicks
2. configureSimulation function gets generated as a new tab. You may need to copy code from here and paste it in other document.



# Generating CC3D C++ extension modules using Twedit++

- Writing plugin or steppable code in C++ is often a mechanistic process – at least when you being. It can be also very tedious
- Up to 10 files need to be generated or modified
- Twedit++ autogenerates a **working** plugin or steppable for you freeing you from error-prone manual process
- Twedit++ has C++ code assistant which is not as comprehensive as Python or CC3DML assistants but we can certainly change it. Your input is greatly appreciated
- C++ Code Generated in Twedit can be then edited in any editor you wish including Twedit++

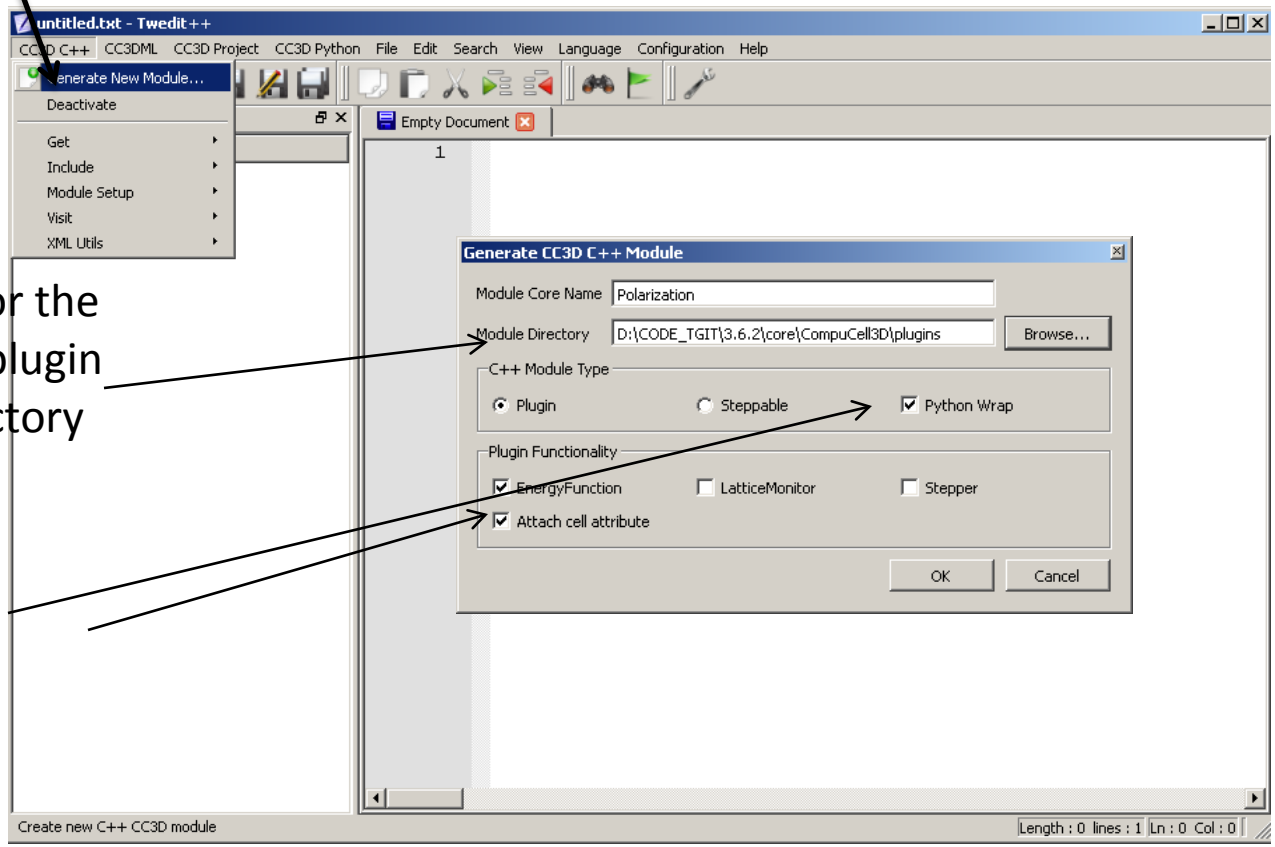
# Generating CC3D C++ extension module

After clicking OK on the Generate CC3D C++ Module dialog box Twedit++ will generate/modify about 10 files and open them as separate tabs.

Go to CC3D C++ menu and choose Generate New Module

Pick appropriate directory for the module – here we develop plugin so we pick ‘plugins’ subdirectory in the CC3D source tree

You can attach extra cell attributes and make your module Python callable



# Autogenerated CC3D C++ plugin code