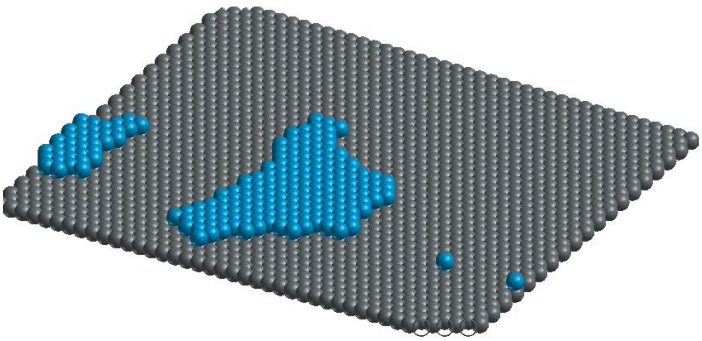
Innovative Coating Solutions,
Place Saint-Pierre, 11, B-5380 Forville., Belgium
slu@incosol4u.com www.incosol4u.com

VIRTUAL COATER™

Virtual Coater GUI Manual for version 4.8

A. Fauroux, J. Müller, P. Moskovkin, S. Lucas



University of Namur (UNamur)

Namur Institute for Structured Materials (NISM)

Laboratoire d'Analyses par Réactions Nucléaires (LARN)

Innovative Coating Solutions (ICS)

https://www.incosol4u.com/nascam-general

Innovative Coating Solutions,
Place Saint-Pierre, 11, B-5380 Forville., Belgium
slu@incosol4u.com www.incosol4u.com

Table of Content

Innovative Coating Solutions,
Place Saint-Pierre, 11, B-5380 Forville., Belgium
slu@incosol4u.com www.incosol4u.com

1. Welcome to the Virtual Coater user interface manual

Welcome to the *Virtual Coater* user interface manual: a user interface written in Java for Virtual Coater:

Virtual Coater (Nanoscale Modelling) is a 2D-3D Kinetic Monte Carlo code for the simulation of deposition, diffusion, nucleation and growth of films on a surface, developed in Namur University. Please consult the documentation or consult the university web page for more information on Virtual Coater itself.

The aim of *Virtual Coater* is to provide a clear and easy to use interface for simulating deposition, diffusion, nucleation and growth of films on a surface as well as to provide an easy way for other scientists to implements plugins that will analyse the films produced during those simulations, or that will produce customized input parameters.

Have fun,

Prof. S. Lucas

1.1 Description

- Software: Virtual Coater user interface
- Platform: Windows
- Author: Antoine Fauroux antoine.fauroux@unamur.be
- Language: Java (minimum jre version 1.6)
- Libraries:
  - jfreechart-1.0.19 LGPL
  - jmol_14.4.4_2016.03.31 LGPL
  - sqlite-jdbc-3.8.11.2 LGPL
  - pdfbox-app-2.0.2 Apache License, Version 2.0
  - opencsv-3.8.jar Apache License, Version 2.0

Innovative Coating Solutions,
Place Saint-Pierre, 11, B-5380 Forville., Belgium
slu@incosol4u.com www.incosol4u.com

2. Getting Started

2.1 Installation

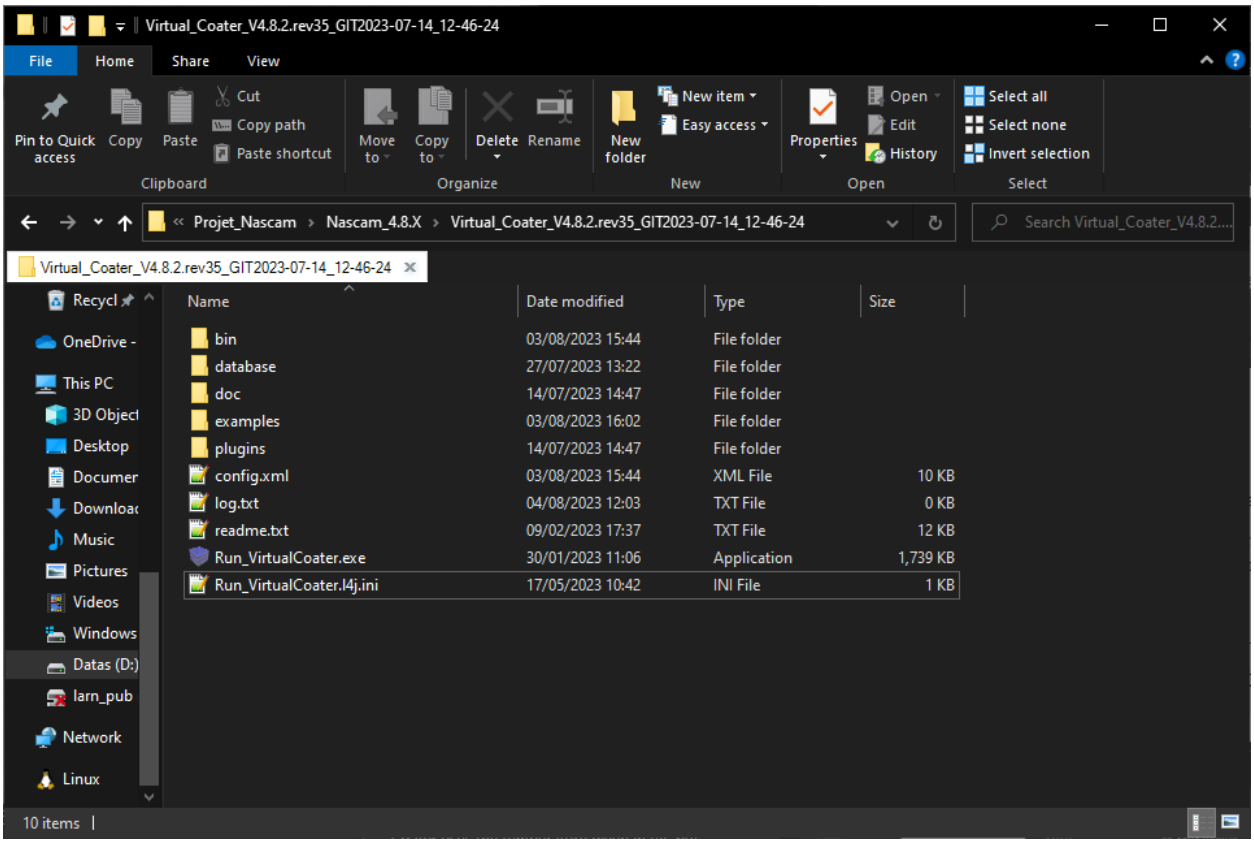
Virtual Coater is a standalone software. This mean that you just need to download the latest version archive, and to decompress it on your computer, and you will be able to run it from where you extracted it.

2.2 Running

To run *Virtual Coater* you just have to click on the *Run_VirtualCoater.exe* executable at the root of the extracted folder.

This software was written in Java and is using a wrapper to launch VirtualCoater.exe; you can pass if you want arguments to the Java Virtual Machine (JVM) if you enter them in the file **Run_VirtualCoater.l4j.ini**. By default the allocated Heap size minimal and maximal memory are set to a quarter of your RAM. If you have lot of RAM, you can put higher values and this will improve for example the loading time for big structures in the structure viewer.

If you have a 32 bit Java engine installed on your computer you might not be able to allocate more than 1200 Mb to the JVM.

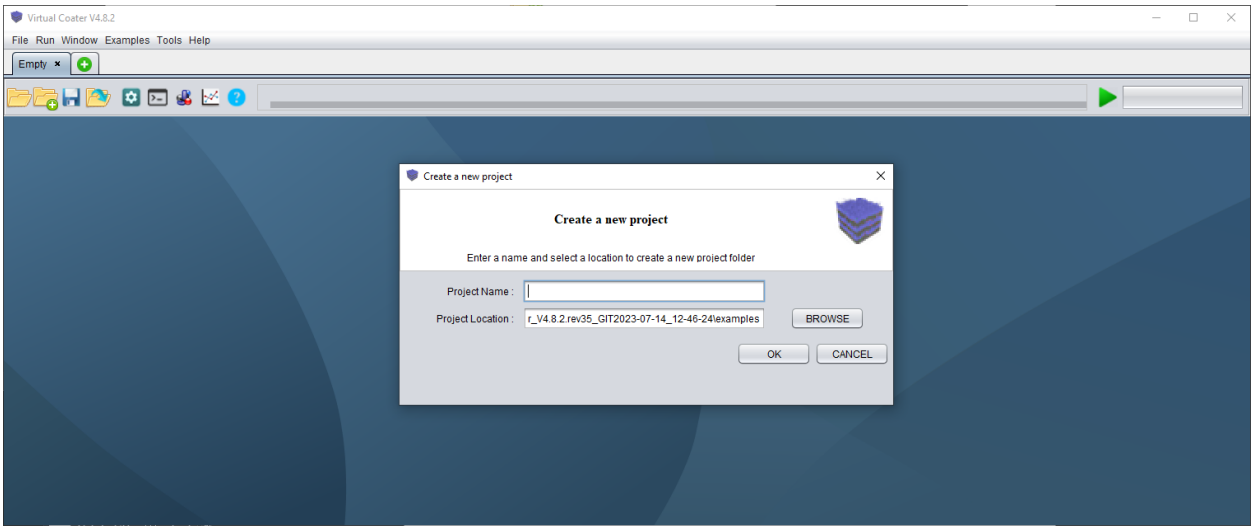


Innovative Coating Solutions,
Place Saint-Pierre, 11, B-5380 Forville., Belgium
slu@incosol4u.com www.incosol4u.com

2.3 Create a New Project

To create a new project with default settings click on the  icon or alternatively got to the menu **File>New Project.**

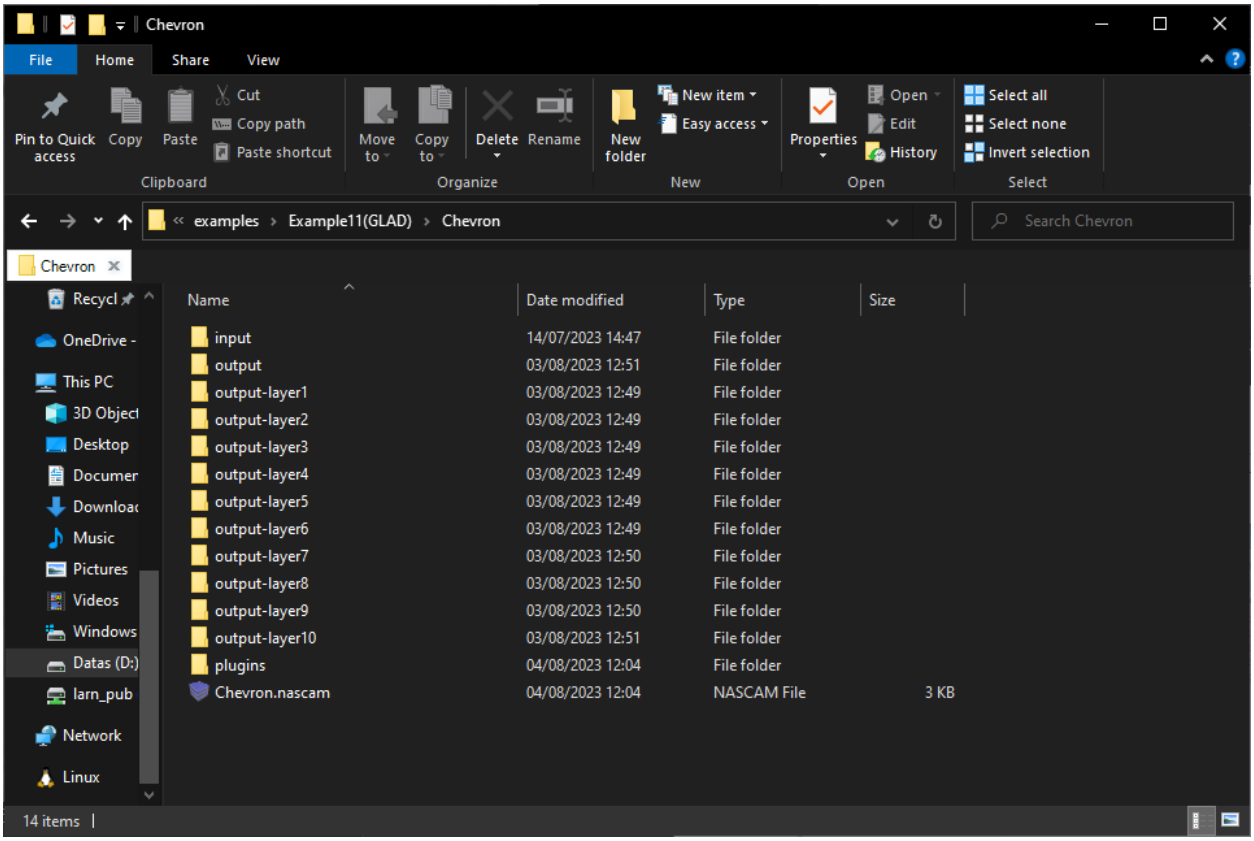
You will be presented with a window where you will have to select the location and the name of your new project.



2.4 The Project Folder

For each new simulation you have to create a *project* or modify an existing one. A new *project folder* will be created by the GUI with the following structure:

Innovative Coating Solutions,
Place Saint-Pierre, 11, B-5380 Forville., Belgium
slu@incosol4u.com www.incosol4u.com



This project folder contains:

- an *input* folder that will contain the simulation input files.

- an *output* folder that will contain the simulation outputs after the simulation will be run.

- a *plugins* folder that will contain all the plugins inputs and outputs.

- a project save file with a *.nascam* extension. The reason for the use of *.nascam* extension is that if you associate this type of file with Run_VirtualCoater.exe, you will be able to open a project with a double click on this *[yourProject].Virtual Coater* file.

Generating a new project all its folders are empty excepting the input folder that contains a file called input.txt – for more information see below and see the Virtual Coater manual too.

2.5 First Look at the GUI

The Menu is self-explanatory but let me point here at the *Examples* and *Plugins* menu. The Examples menu contains all the examples mentioned in this documentation and provide a good way to understand Virtual Coater capabilities. To have a more detailed description of each example you can refer to the example documentation.

Innovative Coating Solutions,
Place Saint-Pierre, 11, B-5380 Forville., Belgium
slu@incosol4u.com www.incosol4u.com

The *Plugins* menu contains all the available plugins but some other ones can be added easily later on. Please refer to the Plugins Section of this wiki for a comprehensive explanation of the plugin functionalities.

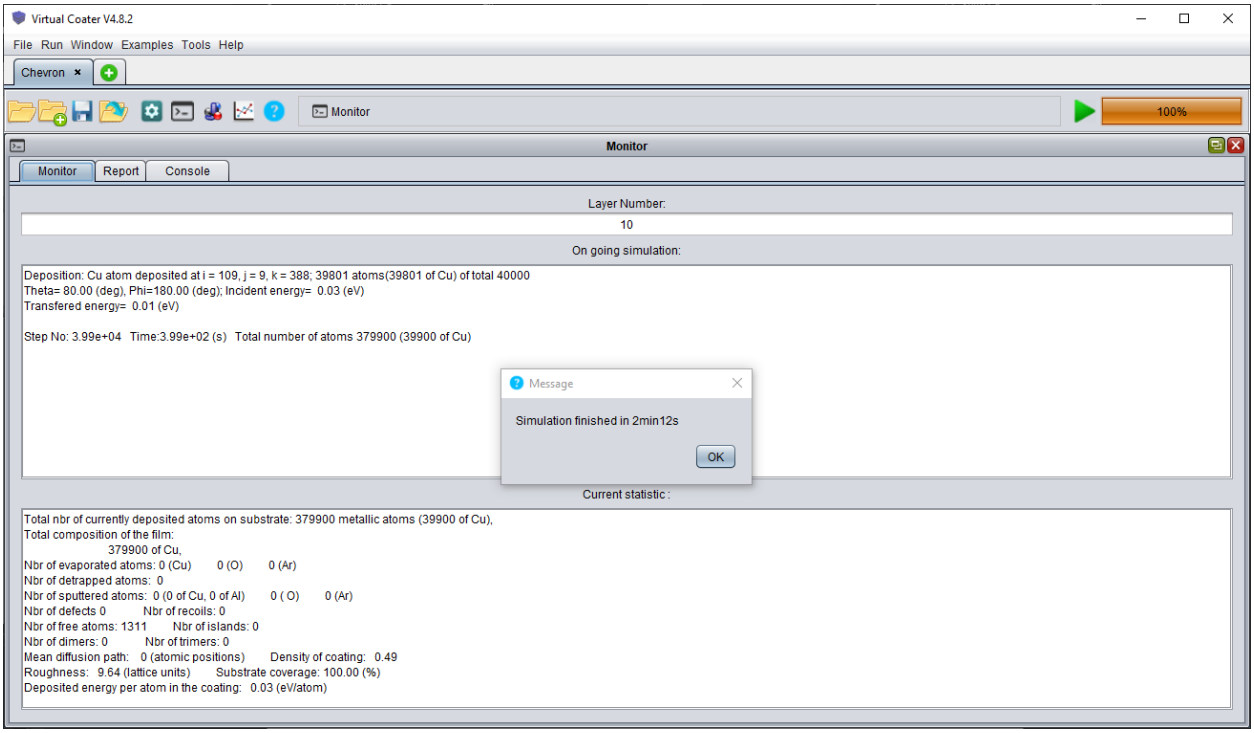
2.5.1 Description of the control bar icons

	*Open an existing project*
	*Create a new project*
	*Save the current project (saves the input parameters of the simulation and the project preferences (?))*
	*Open the project folder in explorer*
	*Open the parameter frame*
	*Open the monitor frame*
	*Open the structure viewer (we use for a structure viewer the Jmol library that have lots of really useful features)*
	*Open the graphics frame. The graphics frame provides lot of statistics on the simulation results.*
	*Open a Help dialog that gives easy access to the Virtual Coater Manuals*
	*Run/Stop the current project simulation*
	*Close the current project*

2.6 Launch your first project

With a new project or an example project opened, press the run button. You should see the Simulation monitor frame popping out and displaying statistics about the ongoing simulation. The progress bar will display the current progression.

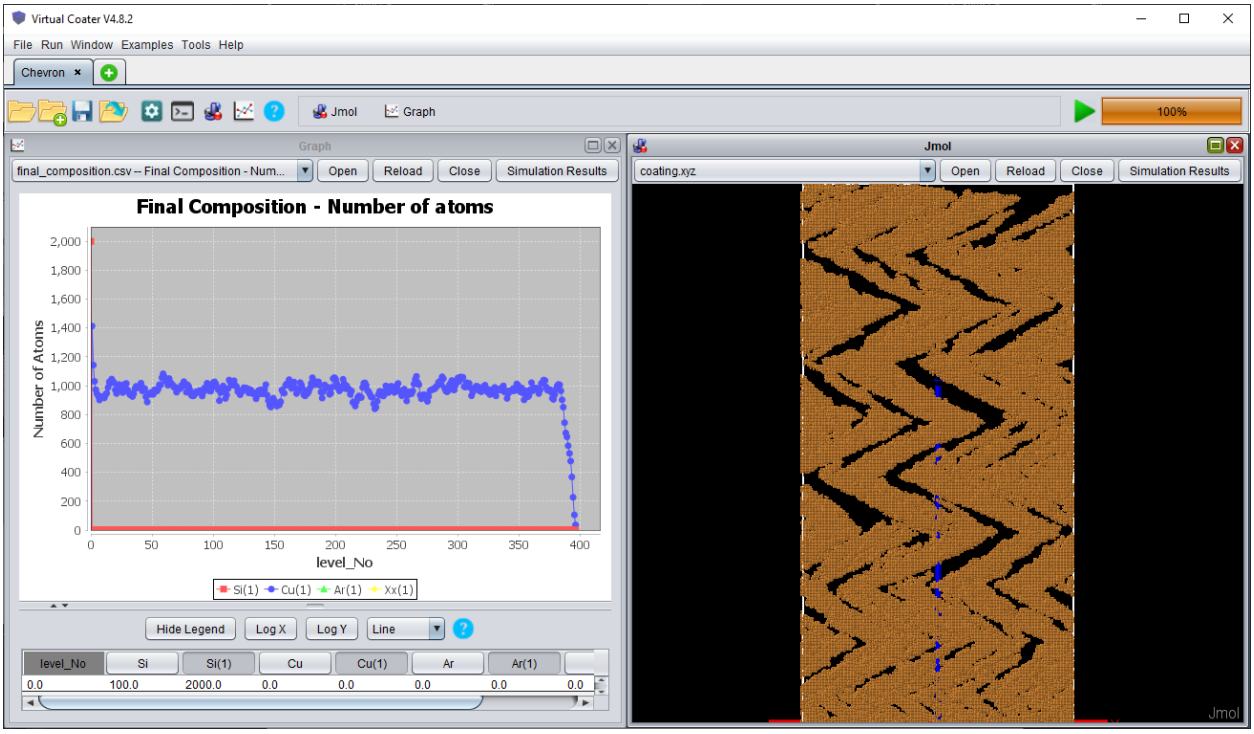
To stop the project at any time you can press the stop icon



2.7 Results

To see if everything went alright you can check out the log in the Monitor and to check the results of the simulation easily you can check the Structure Panel and the Graphics Panel.

Innovative Coating Solutions,
Place Saint-Pierre, 11, B-5380 Forville., Belgium
slu@incosol4u.com www.incosol4u.com



Innovative Coating Solutions,
Place Saint-Pierre, 11, B-5380 Forville., Belgium
slu@incosol4u.com www.incosol4u.com

3. The Internal Frames

3.1 The Parameter Frame

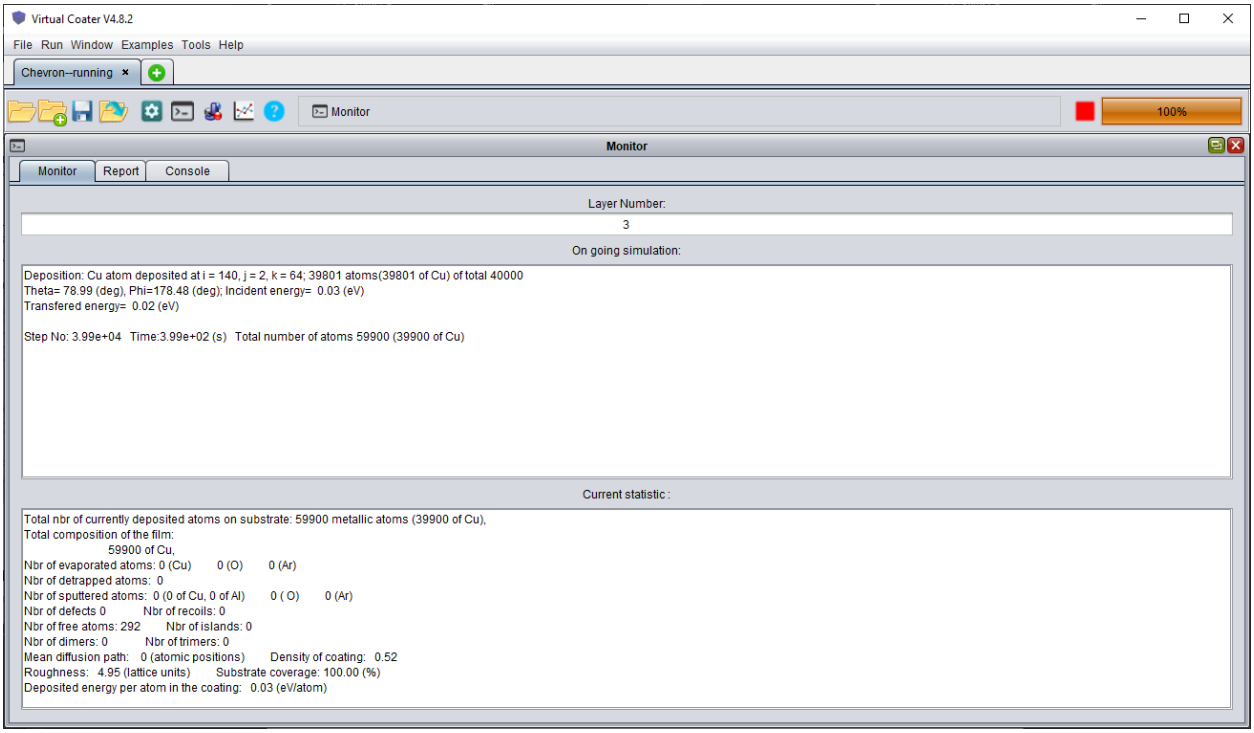
Where you set the simulation parameters (See The Parameter Frame page)

3.2 The Monitor Frame

The monitor help you track what is happening in the background. It is accessible even if no project is loaded and you can find here a **Console** where all the logs are stored.

If you have a problem, you can look at the console or a precise log.

The two other tabs are related only to the current simulation and provide a resume of its current state. The **Monitor** tab displays the current simulation log whereas the **Simulation Report** tab displays the entire simulation log. When a new simulation is launched these two tabs are cleared before the new simulation start.



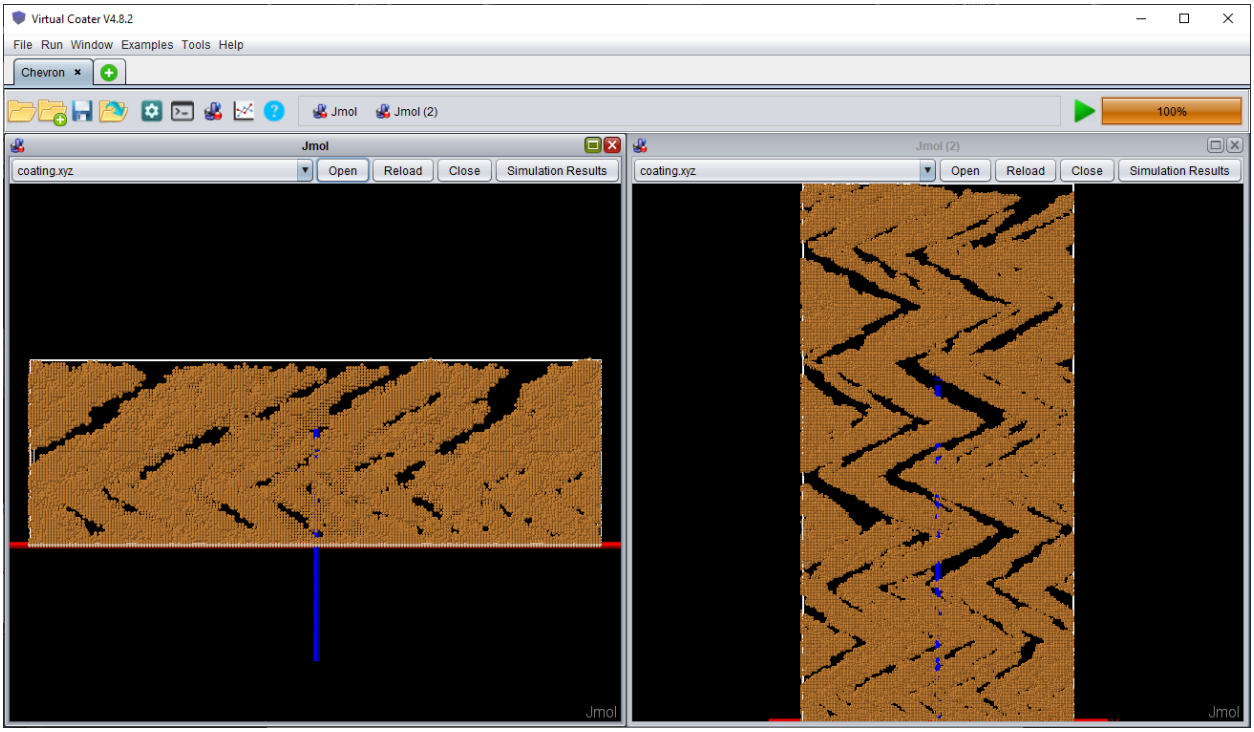
3.3 The Structure Viewer

The **Structure Viewer** is a panel using the Jmol library to display atomic structures and molecules. To know which are the structure files output by the simulation check the Output Files page. With it you can:

- Open any file in the XYZ format

Innovative Coating Solutions,
Place Saint-Pierre, 11, B-5380 Forville., Belgium
slu@incosol4u.com www.incosol4u.com

- Load the final result of the current project (if you have run the simulation at least once)
- With a right click you can access the Jmol pop up menu with lots of options (including imports and exports)
- Open a Jmol console and parse some commands to the viewer. For example, `display x>0` will display only the atoms on the positive side of the X axis. To browse the available commands, go to http://chemapps.stolaf.edu/jmol/docs/
- You can play an animation of the current simulation. Here's how to proceed:
  1. Run a simulation with **Save intermediate 3d structures** checked and set the number of deposited atoms between two saves (**Save intermediate 3d** structures every) to a smaller value than the total number of deposited atoms
  2. Open the file **film_growth.xyz** in the **output** folder (for multi-layer deposition look into the **output** folder)
  3. After a right click on the structure viewer panel select the menu **Animation > Play**

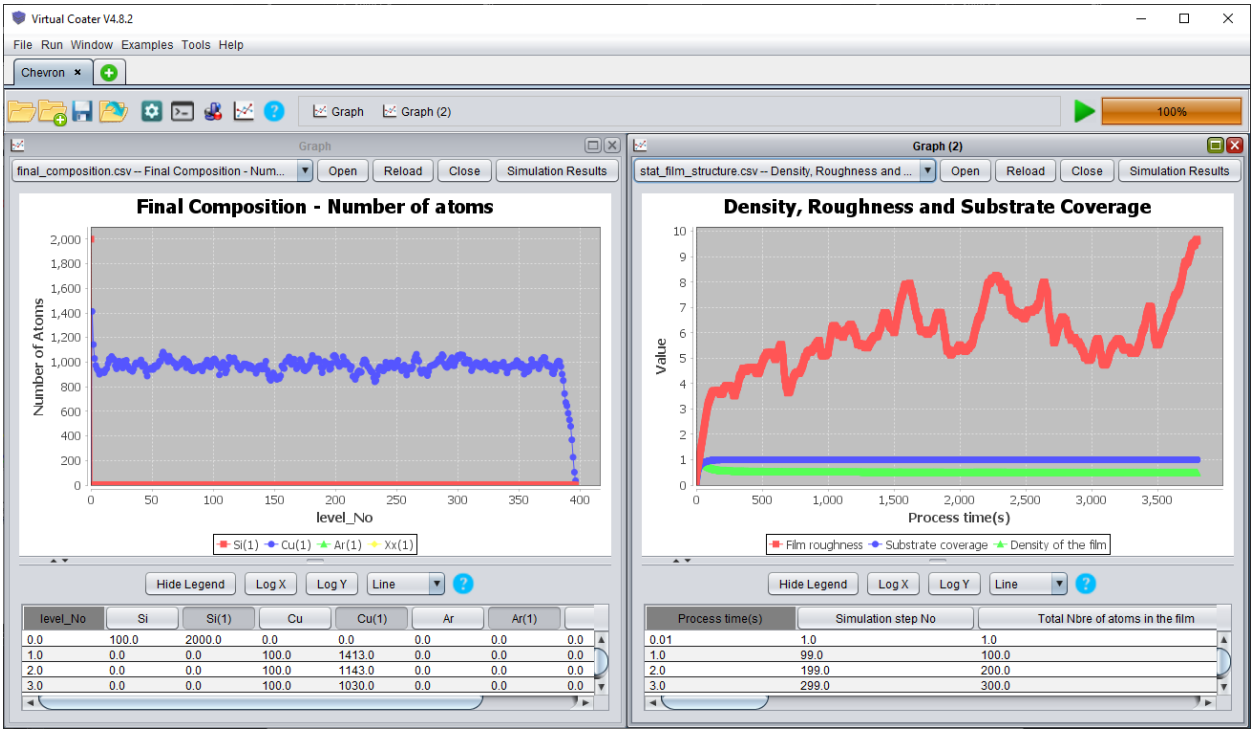


3.4 The Graphics Frame

The **Graphics** frame gives you an easy way to Virtualize the statistics produced during the current simulation. To know more about the statistics output files check the output files page.

*You can change the X axis from **number of deposited atoms** to **simulation time** in the **Settings** menu for better reading.*

Innovative Coating Solutions,
Place Saint-Pierre, 11, B-5380 Forville., Belgium
slu@incosol4u.com www.incosol4u.com



4. The Parameters Frame

The **Parameters** frame contains all the simulation parameters fields as well as some general configuration parameters. It includes lot of panels, tabs and features. A detailed description of each tab is given in the following.

It consists of many tabs with self-explanatory names:

4.1 Model Description

This is a text area giving you the possibility to add a short description of the current project.

- **Input File**

There will be many of such fields with a **[...]** button. Their purpose is to let you easily browse your disk to import some files. Here it is a way to import an input.txt file and load the simulation parameters from it. It is useful if you want to re-use the parameters of another project for example.

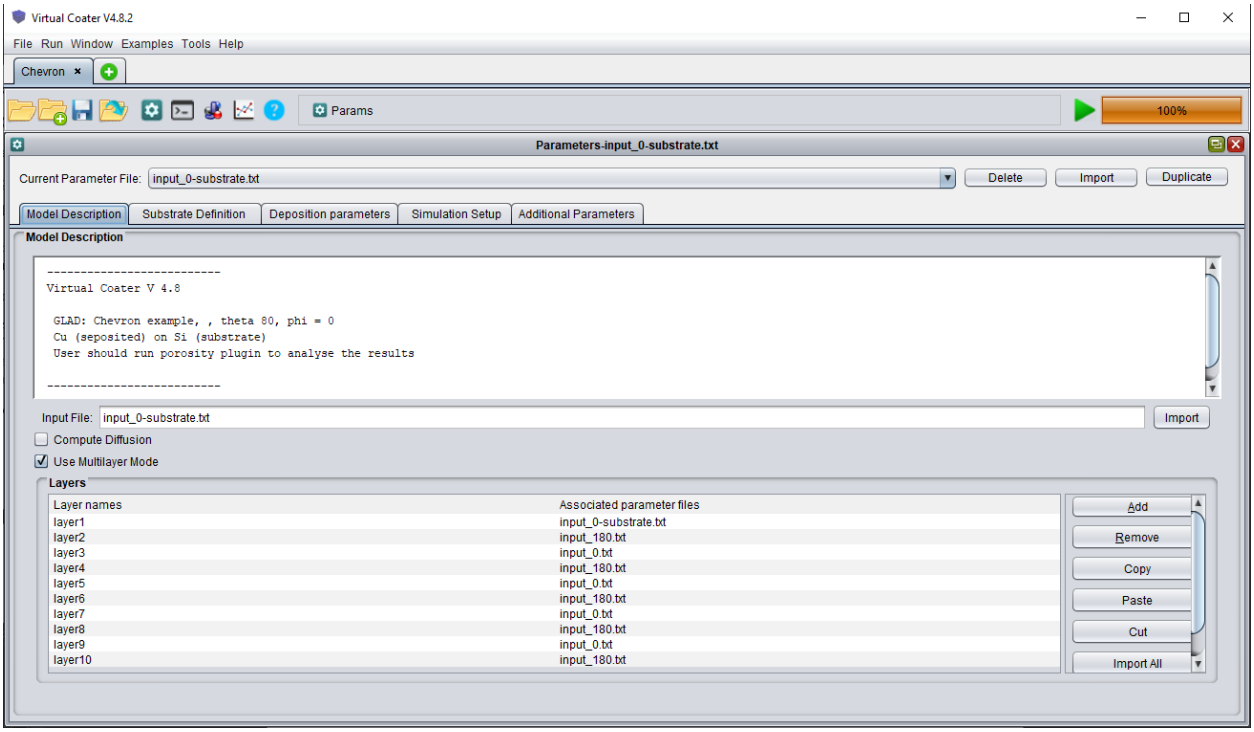
- **Compute Diffusion**

Once checked, all thermal activated events corresponding to the Kinetic Monte-Carlo model used by Virtual Coater are taken into account during the current simulation. A new panel, **Substrate Physical Constant,** is added to the **Parameters** frame and the user have to choose the activation energy value corresponding to each possible event inside.

- **Use Multilayer Mode**

Innovative Coating Solutions,
Place Saint-Pierre, 11, B-5380 Forville., Belgium
slu@incosol4u.com www.incosol4u.com

Gives the possibility to do multi-layered deposition (see Multi-Layer Mode). Checking this option will create a new tab letting you edit different input files and to associate them with each layer.



4.2 Substrate Definition

The second tab of **Parameters** frame is dedicated to the definition of the substrate used by the current simulation. Below are described all its fields.

- **Substrate Viewer and Substrate Loading**

On the top of this tab you can see a structure viewer showing the substrate. If you want to import a custom substrate structure you can use the **[...]** button. The information in the corresponding file must be in the XYZ format.

Sometimes the viewer is not updated and can be misleading, be sure to reload it.

- **Substrate Dimensions**

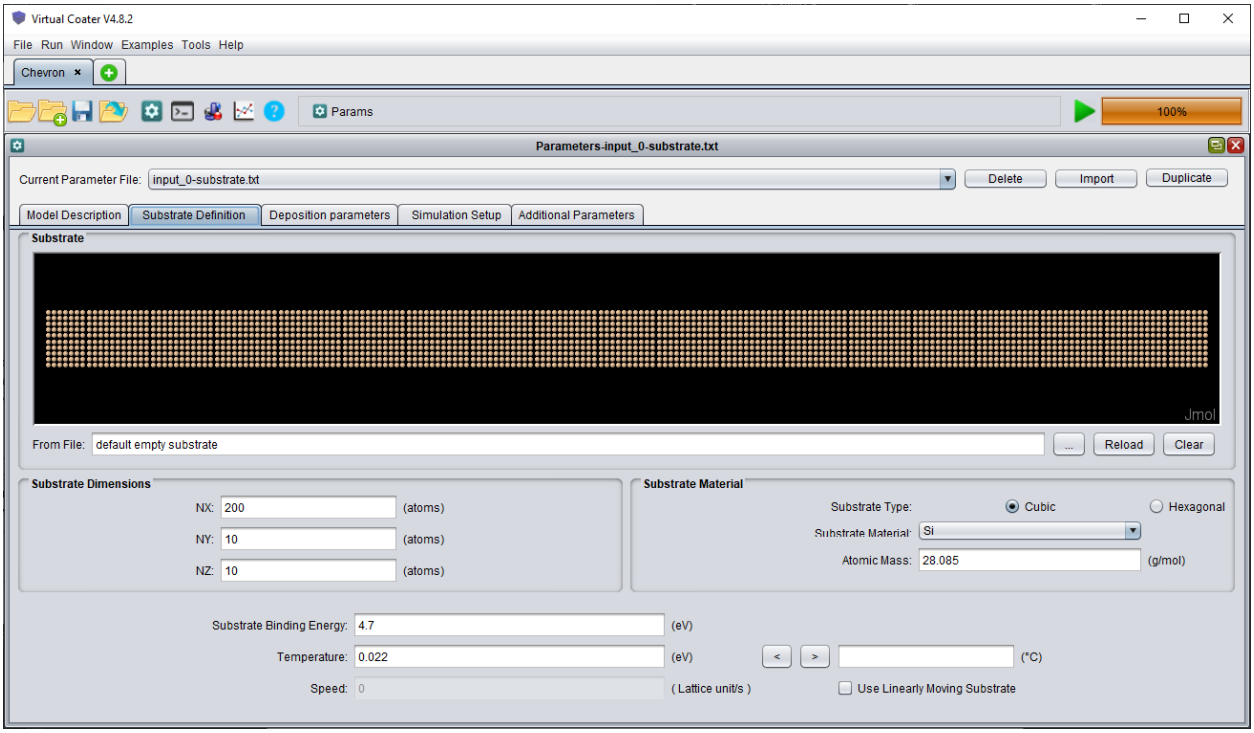
Defines the dimensions of the simulation box. If there is no substrate file loaded then a default empty substrate will be generated. This substrate will be one atom thick all belonging particles having the z coordinate equals to zero.

- **Substrate Material**

Establishes the chemical nature and symmetry of the substrate. A user can choose a chemical element from the available database and the atomic mass will chance correspondingly. In what concerning the substrate symmetry, there are two possible options in the current Virtual Coater version: cubic or hexagonal substrate symmetry.

- Under the same **Substrate Definitions** frame the user can choose the **Substrate Binding Energy** and the current simulation **Temperature**. The first parameter is used to compute the threshold energy for a defect creation. The second one is effective if the user validated the Compute Diffusion option under Model Description frame too. Virtual Coater uses the temperature given in eV, but its value in °C can be checked and even established using the corresponding arrow. Of course, after establishing the temperature right value in eV starting from its value in °C the user must save his choice before to start a new simulation.

- **Use Linearly Moving Mask** option gives the possibility to use a moving mask during the deposition process and to establish its speed (see below).



Innovative Coating Solutions,
Place Saint-Pierre, 11, B-5380 Forville., Belgium
slu@incosol4u.com www.incosol4u.com

4.3 Deposition Parameters

All information about the geometric and energetic characteristics of the incoming particles and their relative amount can be established under this frame. Oscillation and rotation movement characteristics of the substrate and its eventual initial tilt angle can be modified here too. If a mask is used during the deposition process, it is defined (or the file containing it is specified) here.

- **Metal/Gas tabs**

These tabs allow defining independently their chemical nature, amount in the deposition flux and angular and energetic properties. Different options are available for the two last items: the user can define them directly or give the filenames of the files containing their corresponding distributions.

If any of the two checkboxes corresponding to **Use Masked Deposition** or **Use Tilt & Rotation** a corresponding tab appears

- **Use Masked Deposition**

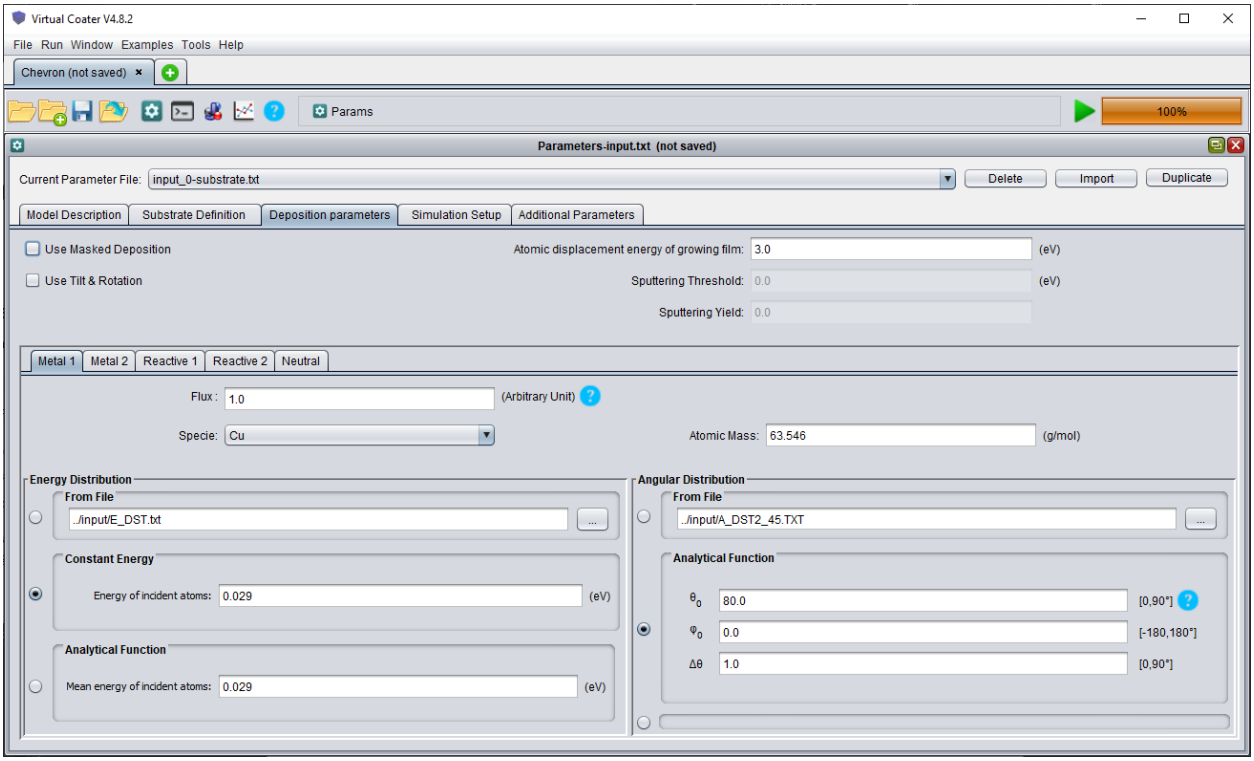
Allows the user to define a deposition mask easily or to import it from a file. If **Use Linearly Moving Mask** has been checked under **Substrate Definitions** frame, the incoming particles will reach the substrate passing through a moving mask. For more details about the mask definition see the Virtual Coater Manual.

*Sometimes the pattern display is not well updated. You should click **Reload** or save the project to be sure that your mask definition has been taken into account.*

- **Use Tilt & Rotation**

As already mentioned, the user can decide the initial orientation of the substrate, and if the substrate is or not animated by an oscillating and/or rotation movement during the deposition process.

Innovative Coating Solutions,
Place Saint-Pierre, 11, B-5380 Forville., Belgium
slu@incosol4u.com www.incosol4u.com



4.4 Simulation Setup

The last frame, **Simulation Setup**, is dedicated to the definition of the *general simulation parameters*.

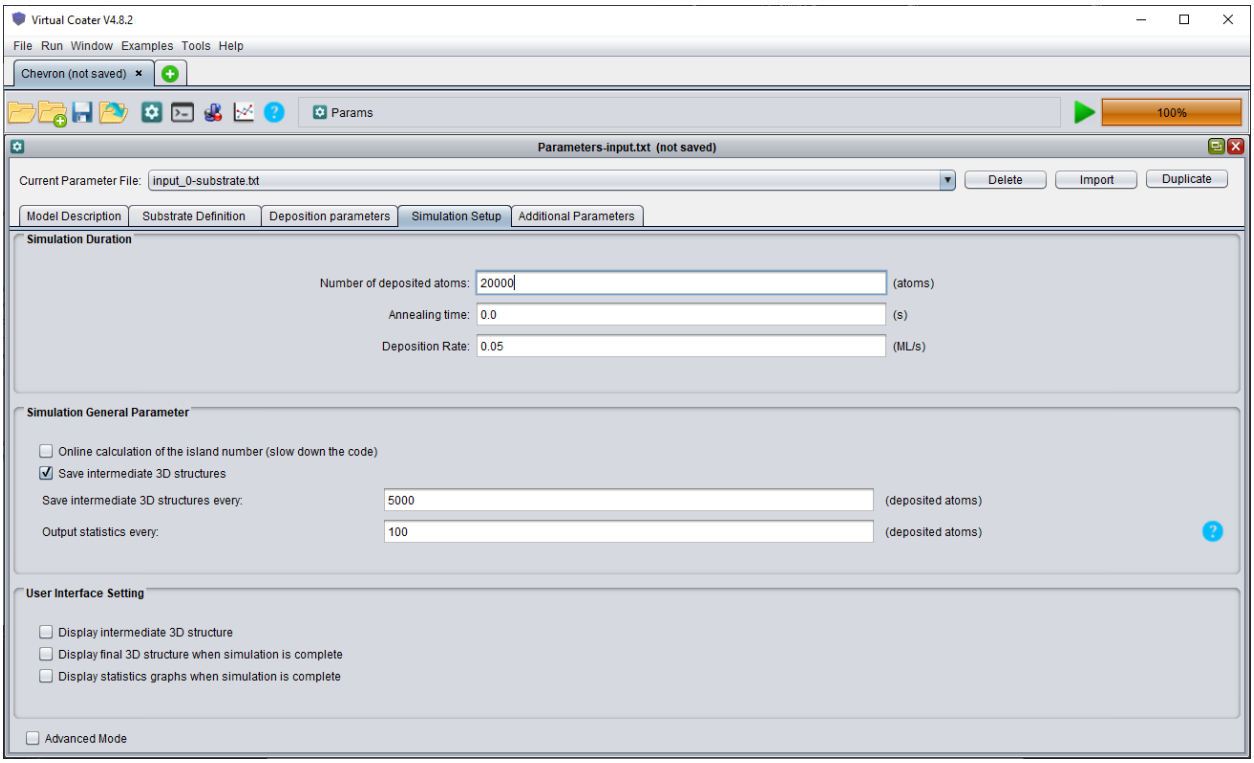
- The **Simulation Duration** tab keeps together some simulation parameters that contribute to the total simulation time. So, the **Deposition Rate** together the **NX** and **NY** parameters (under **Substrate Definition** frame) determine the characteristic time of the system (the time spent between two successive deposition events – see Virtual Coater Manual). The **Number of Deposited Atoms** multiplied by the characteristic time gives the total simulation time under deposition conditions. Obviously, the **Annealing Time** refers to the total simulation time under annealing conditions.

- **General Simulation Parameters** allows the user to decide the output frequencies of saving snapshots of the growing system structure (**Save Intermediate 3D Structures**) and of collecting statistics information about the growing system (**Output Statistics Every**). The update frequency of the simulation progress bar and the **Simulation Monitor** frame is depending on the value chosen for the **Save Statistics Every** tab: lower **Save Statistics Every** chosen value, higher the update frequency for the progress bar and **Simulation Monitor** frame. The island statistics computation can be decided independently checking or not the **No On-Line Calculation of the Island Number**. If this option switched on (unchecked!) then such analysis happens with the same frequency as the other statistics computations (surface roughness, density…).

Innovative Coating Solutions,
Place Saint-Pierre, 11, B-5380 Forville., Belgium
slu@incosol4u.com www.incosol4u.com

- **User Interface Settings** refers to the graphical representation of structural snapshots and/or statistical analysis results during or at the end of the current simulation. The options' names are self-explanatory and we add only that setting **Display intermediate structure** to a high frequency (low value), loading all the intermediate structures might be very resource demanding. A larger working memory for the java virtual machine can be set in the "Virtual Coater.l4j.ini" configuration file.

- **Advanced Mode** displays an additional panel allowing the user to modify two other simulation parameters: **Forced Deposition** and **Prefactor Correction** (see the Virtual Coater Manual for more explanations on their actions).

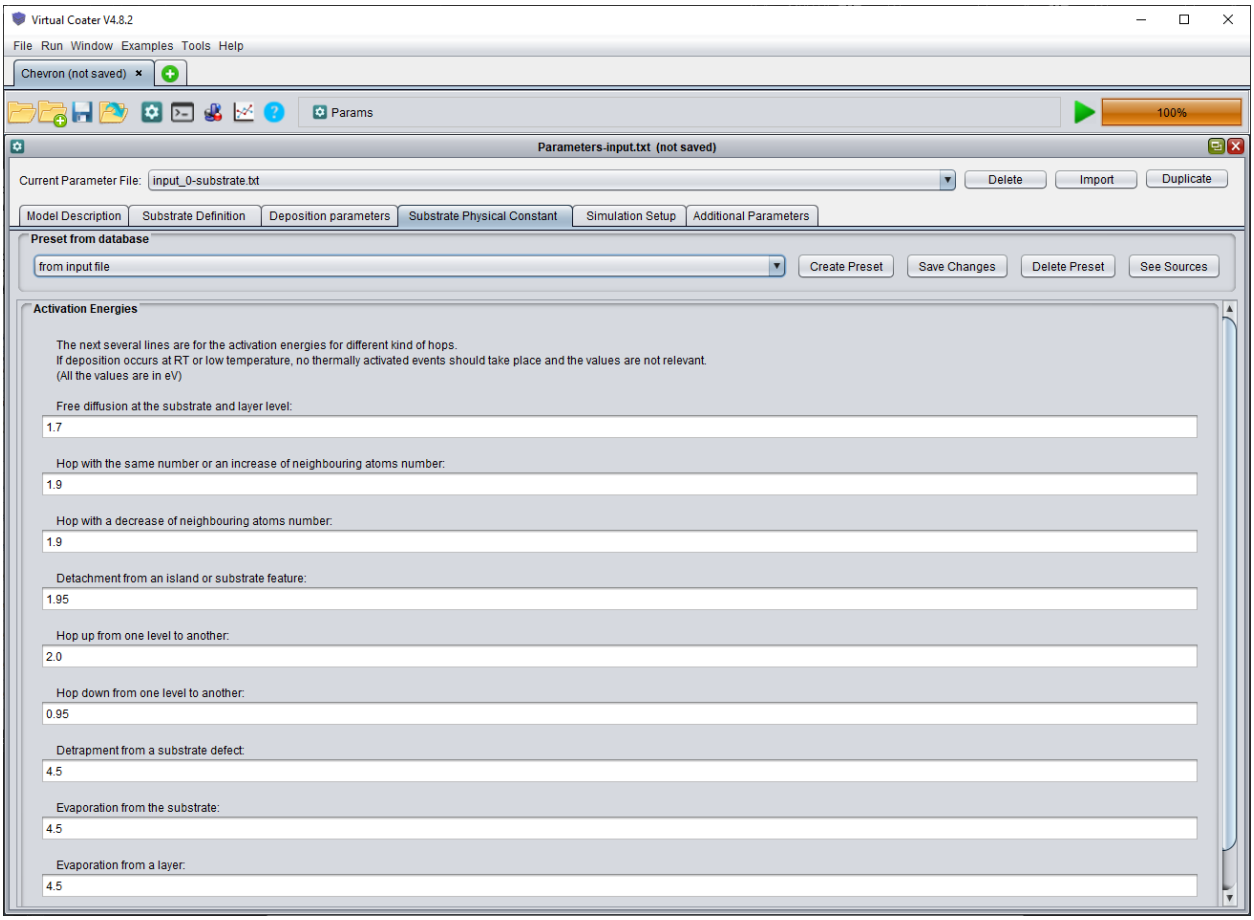


4.5 Substrate Physical Constant

This frame is only accessible if you check **Compute Diffusion** on the first tab (**Model Description**). By default its single tab, **Activation Energies**, contains the activation energies specified in the input.txt file (the single file existing in **input** folder - see above), but you can also load and create presets in the database More information about the meaning of their activation energies can be found in the Virtual Coater manual. To select a preset just use the list **Preset from Database**. To create a preset, firstly modify the energy values and then press **Create Preset** and give a name to the new created preset. By default, the new preset name is the concatenation of the chemical symbols of the species used in the current simulation.

If you save changes onto an existing preset, you will permanently modify it in the database.

You should do a copy of a preset if you want to modify it and to keep the original values intact.



4.6 The Multi-Layer Mode

The multi-Layer mode let you have separate input files and attribute those inputs to different layers.

You can only load a custom substrate file for the first layer

Once you clicked on **Use Multi-Layer Mode** you should:

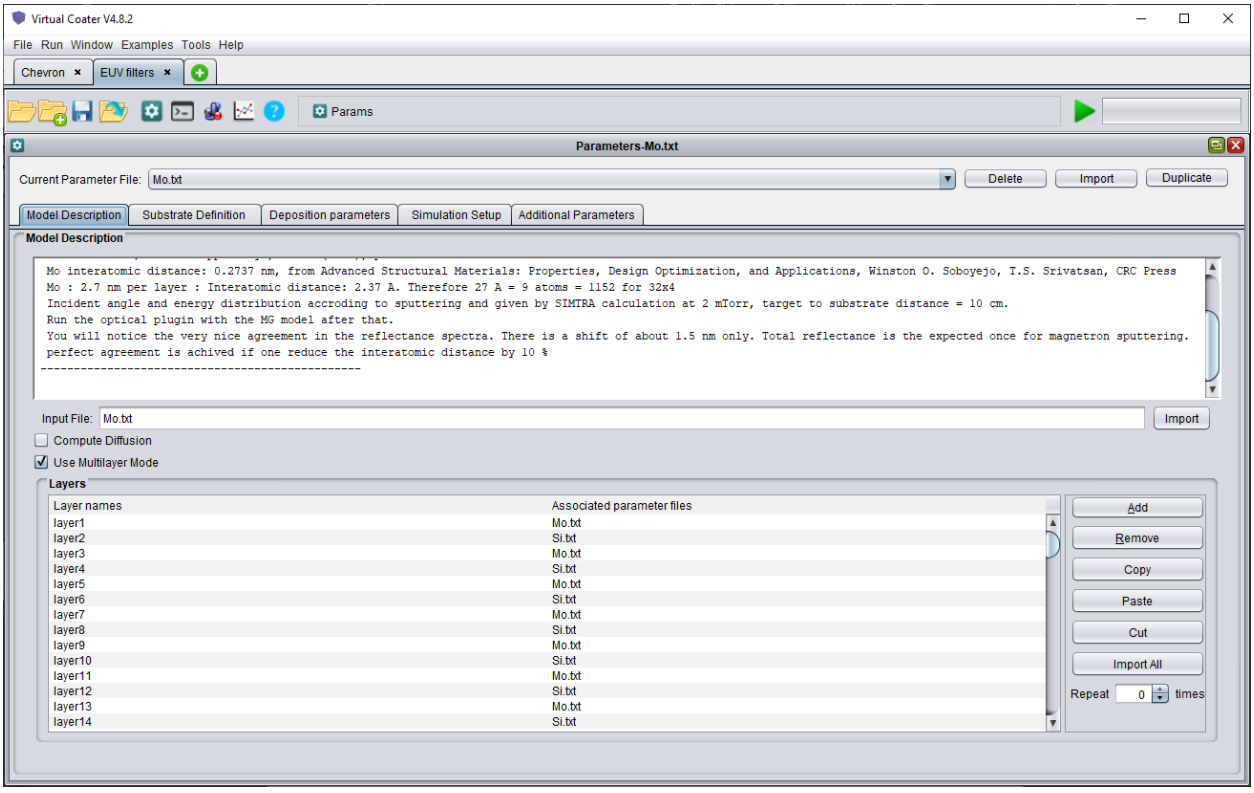
- Create a new input file by pressing on the button **Create New Input** (it will be a copy of the current one with a different name)
- Add some layers by pressing **Add**
- Remove some layers

- Copy and paste layers (they will be inserted at the selected line)
- Modify the input parameters of all the input file corresponding to those layers according to the deposition you want to simulate

- Set a repeat number:

If for example you want to do 120 layers deposition of alternating silicon and molybdenum like in example 12(EUV filter). you set it as:

- layer1 -> Mo.txt # Parameter file for the molybdenum deposition
- layer2 -> Si.txt # Parameter file for the silicium deposition
- Repeated 60 times



To pass from one input file to the other use the list on top of the frame. The name of the current input file you hare modifying is written on the title of the frame.

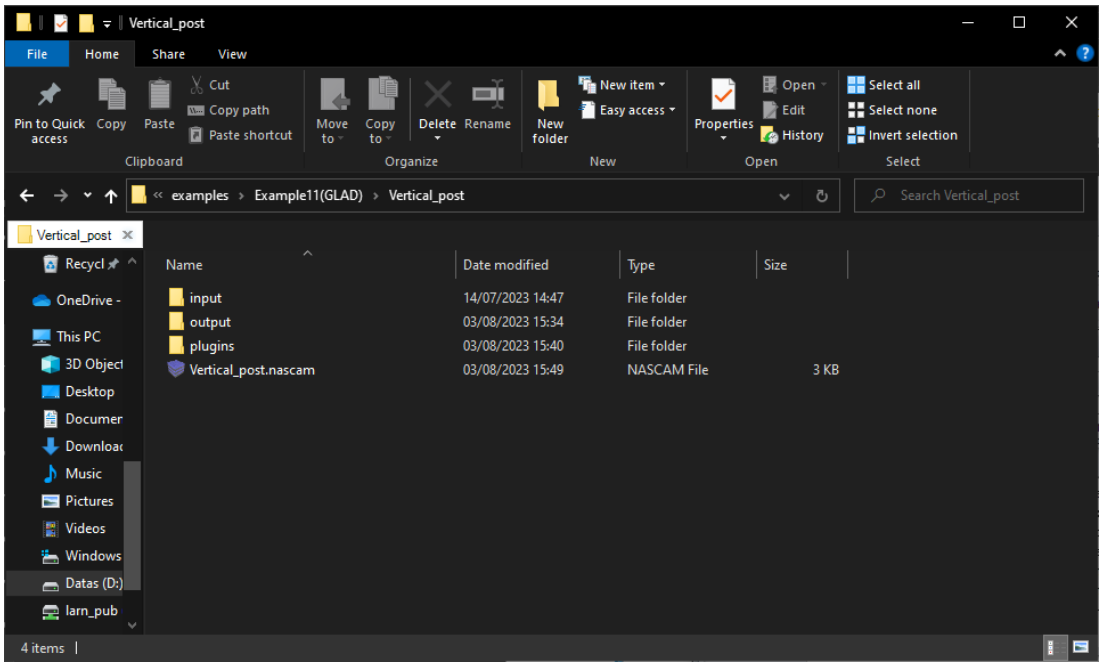
Be sure to save your modifications before changing the current input. Beware, it is possible to run into problem if the parameters for different layer are not compatible. For example, if you set the substrate size to be 20x20 atom wide for the first layer and 20x20 atom wide for the second you risk getting into some troubles

Innovative Coating Solutions,
Place Saint-Pierre, 11, B-5380 Forville., Belgium
slu@incosol4u.com www.incosol4u.com

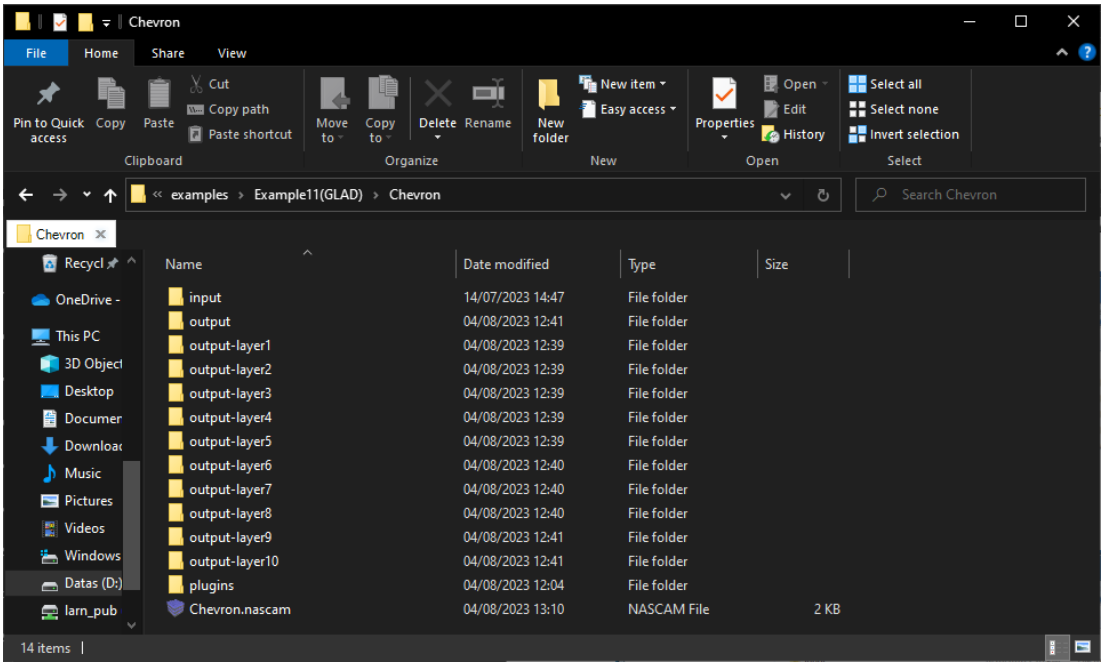
5. Output Files

5.1 The Folder structure

Usually, a project folder contains all the simulation outputs in the **output** folder, and look like that:



But in a case of a multi-layer project, it looks like that:



Innovative Coating Solutions,
Place Saint-Pierre, 11, B-5380 Forville., Belgium
slu@incosol4u.com www.incosol4u.com

And you can find the output files of each layers in the folder **output-layer-{layer number}** and the concatenated version of the statistics and structures of the all simulation in **output** folder.

5.2 The Structure Files

- *film_growth.xyz*

Contains all the intermediate 3D structures

- *coating.xyz*

Contains all the structure at the end of the simulation. In multi-layer mode the coating is the state after each layer and the coating file in the **output** folder contains the final result.

5.3 The Log file

- *log_file.txt*

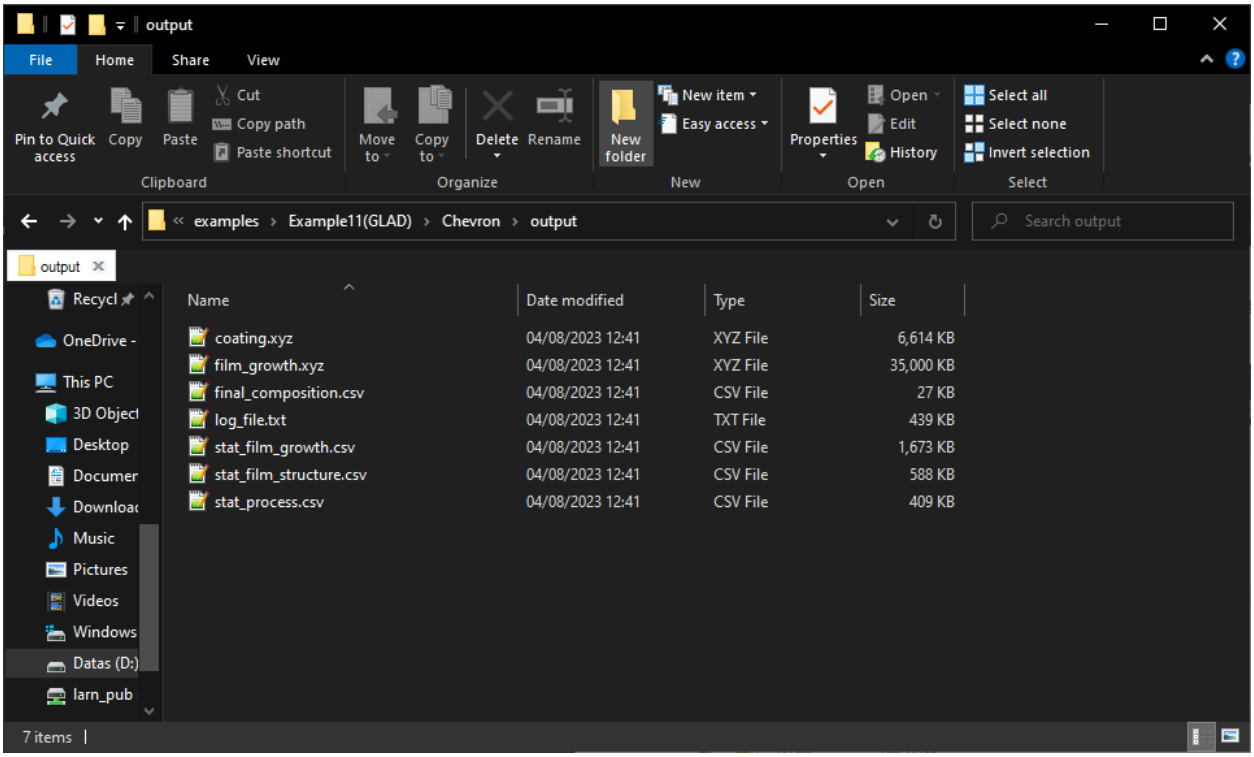
Contains the log of the simulation.

5.4 The Statistics

The statistic files produced by Virtual Coater, consult the [Virtual Coater documentation] for a better explanation. The graphics frame uses those files to produces several graphics that we think are the most relevant, but if you want to Virtualize them with your own way, their format is a standard.

- *stat_process.csv*
- *stat_film_structure.csv*
- *stat_film_growth.csv*
- *Island_distribution.csv*
- *final_composition.csv*

Innovative Coating Solutions,
Place Saint-Pierre, 11, B-5380 Forville., Belgium
slu@incosol4u.com www.incosol4u.com



Innovative Coating Solutions,
Place Saint-Pierre, 11, B-5380 Forville., Belgium
slu@incosol4u.com www.incosol4u.com

6. File Formats

6.1 The Project Save File

The project save file is an XML (Extensible Markup Language) file with a *.nascam* extension. It contains different kind of preferences and general parameters for the associated project. Although it can be edited manually you should better not modify this file as some bad modification can prevent you from opening the project.

The reason for using a *.Virtual Coater* extension is that if you associate this type of file with *Run_Virtual Coater.exe*, you will be able to open a project with a double click on this *[yourProject].Virtual Coater* file.

➢ Example of a project save file:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<projectSave versionNascam="4.8.2">
    <internalFrame type="PARAM">
        <open>true</open>
        <posX>30</posX>
        <posY>30</posY>
        <width>850</width>
        <height>750</height>
        <tab>1</tab>
    </internalFrame>
    <internalFrame type="MONITOR">
        <open>false</open>
        <posX>20</posX>
        <posY>20</posY>
        <width>600</width>
        <height>650</height>
        <tab>0</tab>
    </internalFrame>
    <internalFrame type="JMOL">
        <open>false</open>
        <posX>10</posX>
        <posY>10</posY>
        <width>700</width>
        <height>700</height>
        <tab>0</tab>
        <lastFile></lastFile>
    </internalFrame>
    <inputs>
        <input>input.txt</input>
        <current>input.txt</current>
    </inputs>
    <layers>
        <layer>input.txt</layer>
    </layers>
```

Innovative Coating Solutions,
Place Saint-Pierre, 11, B-5380 Forville., Belgium
slu@incosol4u.com www.incosol4u.com

```
    <settings>
        <useGas>false</useGas>
        <maskedDepo>false</maskedDepo>
        <useTilt>false</useTilt>
        <advanced>false</advanced>
        <openJMol>false</openJMol>
        <openFbox>false</openFbox>
        <openGraphs>false</openGraphs>
        <diffusion>false</diffusion>
        <statisticOut>1</statisticOut>
        <noIsland>false</noIsland>
        <multiLayer>false</multiLayer>
        <substrate>null</substrate>
    </settings>
    <activationEnergiePreset>default</activationEnergiePreset>
</projectSave>
```

6.2 The General Parameters Save File

The General Parameters Save File is a file named **config.xml** located at the root of the Virtual Coater folder. It is created at the first execution of the program and is used to save the general settings, like window size and recent opened project.

You can edit this file manually. For example, you can add some **jmol** commands that will be executed on Jmol opening. By default, we choose to use 3 commands that restrain the log output of Jmol and hide the bound between atoms.

You can check https://chemapps.stolaf.edu/jmol/docs/ for a list of available commands.

If you have troubles opening Virtual Coater you can suppress this file and see if this action solves them.

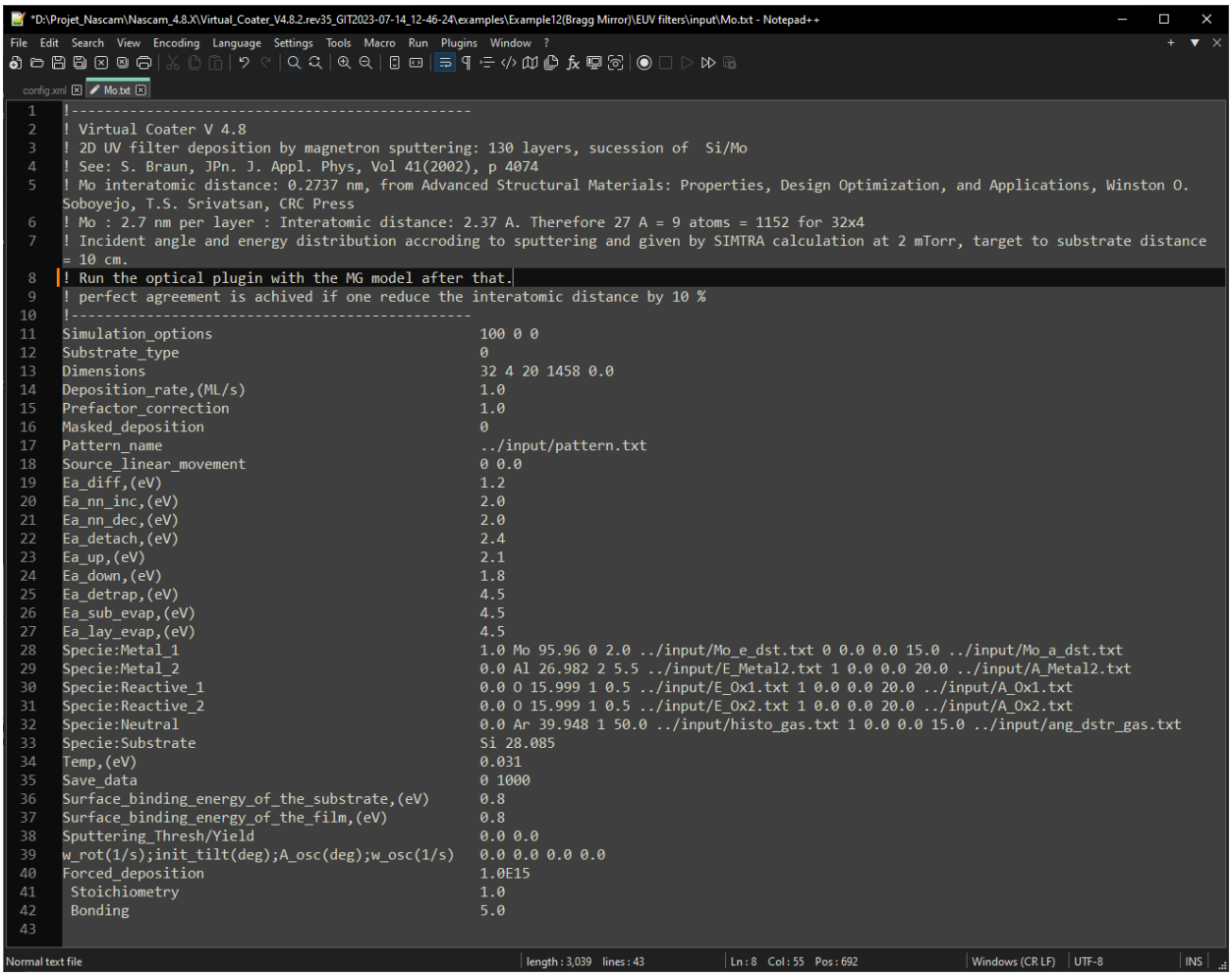
```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<config>
    <version>4.8.2</version>
    <executable>Nascam.exe</executable>
    <extension>.nascam</extension>
    <width>1296</width>
    <height>1000</height>
    <maxRecentProject>6</maxRecentProject>
    <jmolConfig>set autobond off</jmolConfig>
    <jmolConfig>set scriptReportingLevel -1</jmolConfig>
    <jmolConfig>set logLevel 3</jmolConfig>
    <recentProject name="test1" path="C:\Users\afauroux\Desktop\Projects"/>
    <debug>false</debug>
</config>
```

Innovative Coating Solutions,
Place Saint-Pierre, 11, B-5380 Forville., Belgium
slu@incosol4u.com www.incosol4u.com

6.3 The Input File

The *input.txt* file contains the Virtual Coater simulation input parameters. Please refer to Virtual Coater documentation for further info on the meaning of the different parameters.

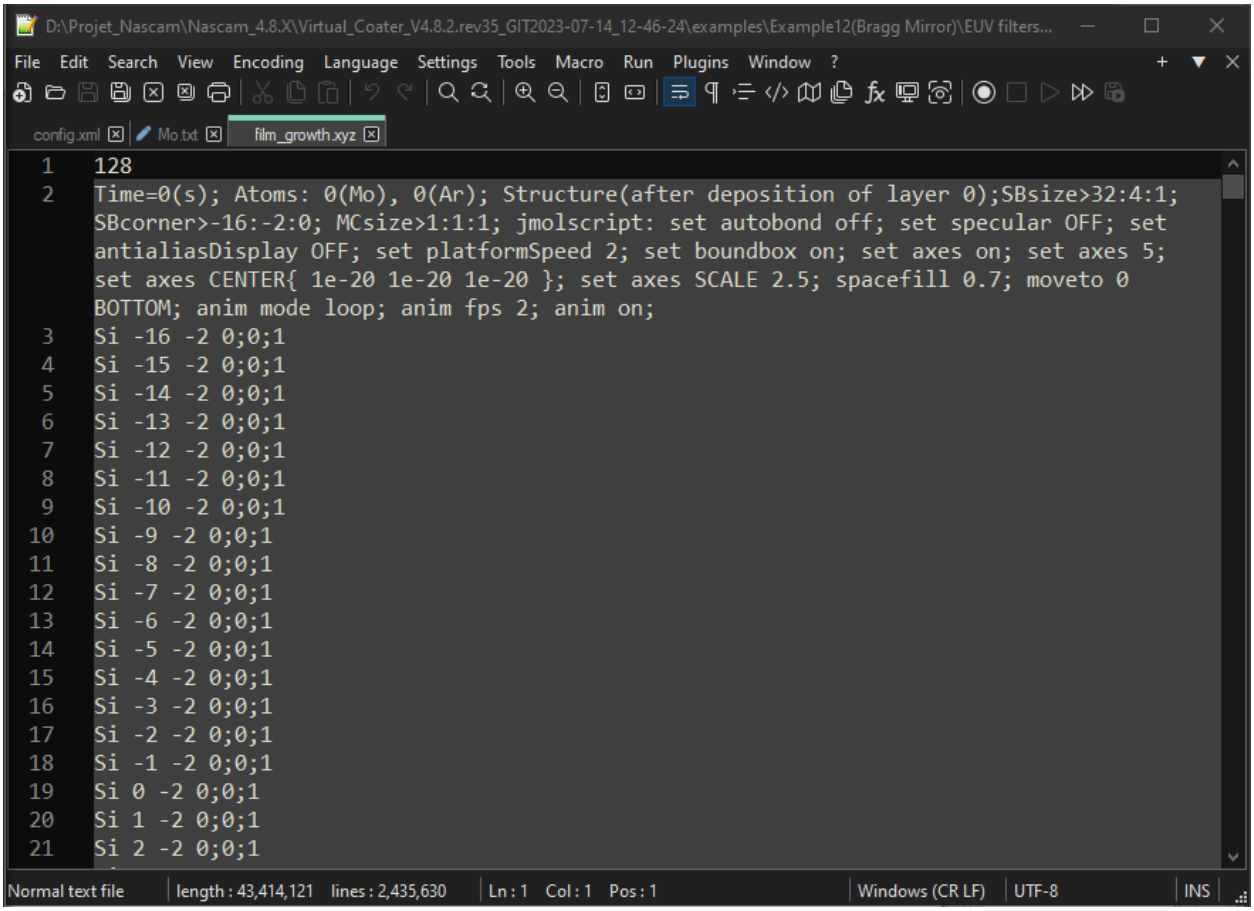
Example of a Virtual Coater input file



6.4 The 3D Structure Files

All files having **XYZ** extension contain information about the system structure, i.e. xyz coordinates and chemical nature of each particle in the system (see XYZ file extension on Jmol wiki). They can also contain several **Models** that are basically several structures in a same file. For example, the file **film_growth.xyz** contains the structure of the coating at various stages of the simulation.

Innovative Coating Solutions,
Place Saint-Pierre, 11, B-5380 Forville., Belgium
slu@incosol4u.com www.incosol4u.com



6.5 The CSV Files

Lots of files have a **CSV** extension. They contain statistics information produced during a simulation or during the execution of a plugin.

By convention the columns are separated by a semicolon, and the first row is reserved to headers – a short description of the corresponding column content.

*For the plugin's outputs, the **CSV** first row is the title of the associated graph preceded with a "!" and the header of the first column is in the format:* ***x axis label / y axis label.***

➢ Simulation output **csv** files:

```
col 1;col 2;col 3;col 4
0;0;0;0
1;0;0;0
2;0;0;0
3;0;0;0
```

➢ Plugins **csv** files:

```
!title of the graph
x axis label/y axis label;col 1;col 2;col 3;col 4
```

Innovative Coating Solutions,
Place Saint-Pierre, 11, B-5380 Forville., Belgium
slu@incosol4u.com www.incosol4u.com

```
0;0;0;0;0
1;0;0;0;0
2;0;0;0;0
3;0;0;0;0
```

Innovative Coating Solutions,
Place Saint-Pierre, 11, B-5380 Forville., Belgium
slu@incosol4u.com www.incosol4u.com

7. The Plugins

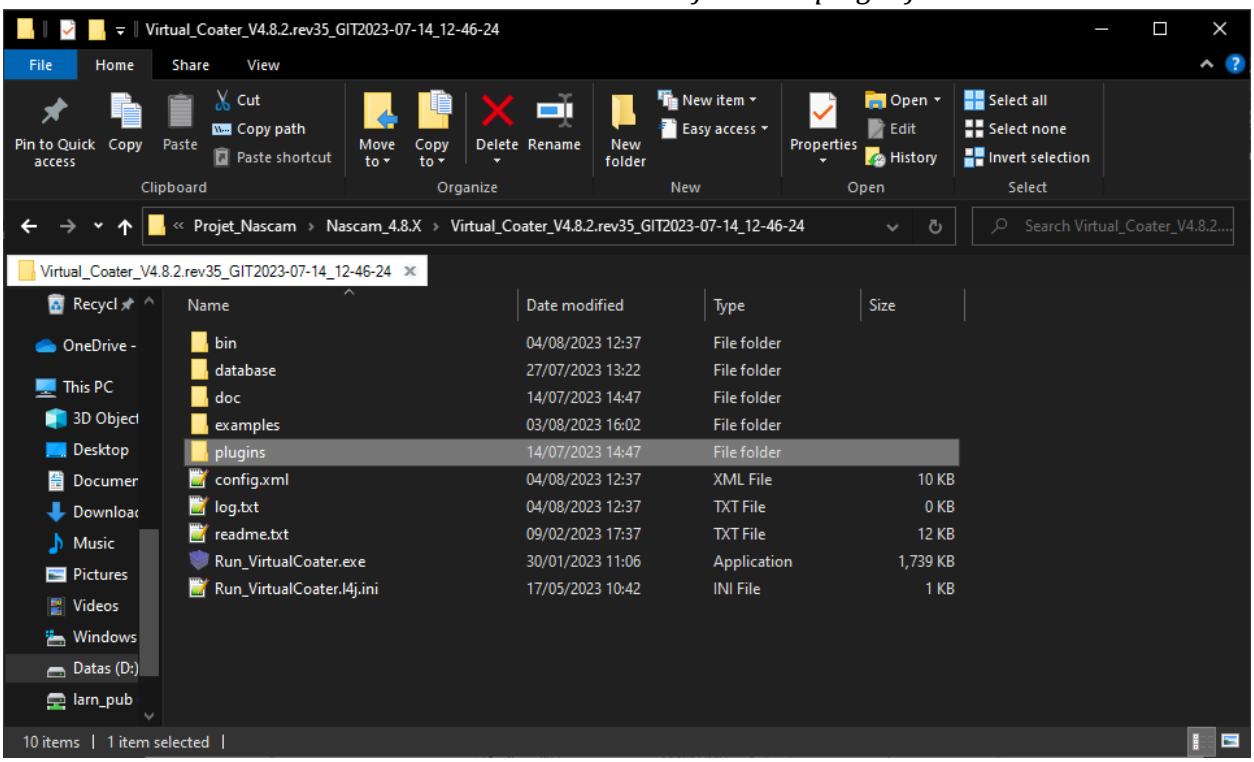
7.1 The plugin folders

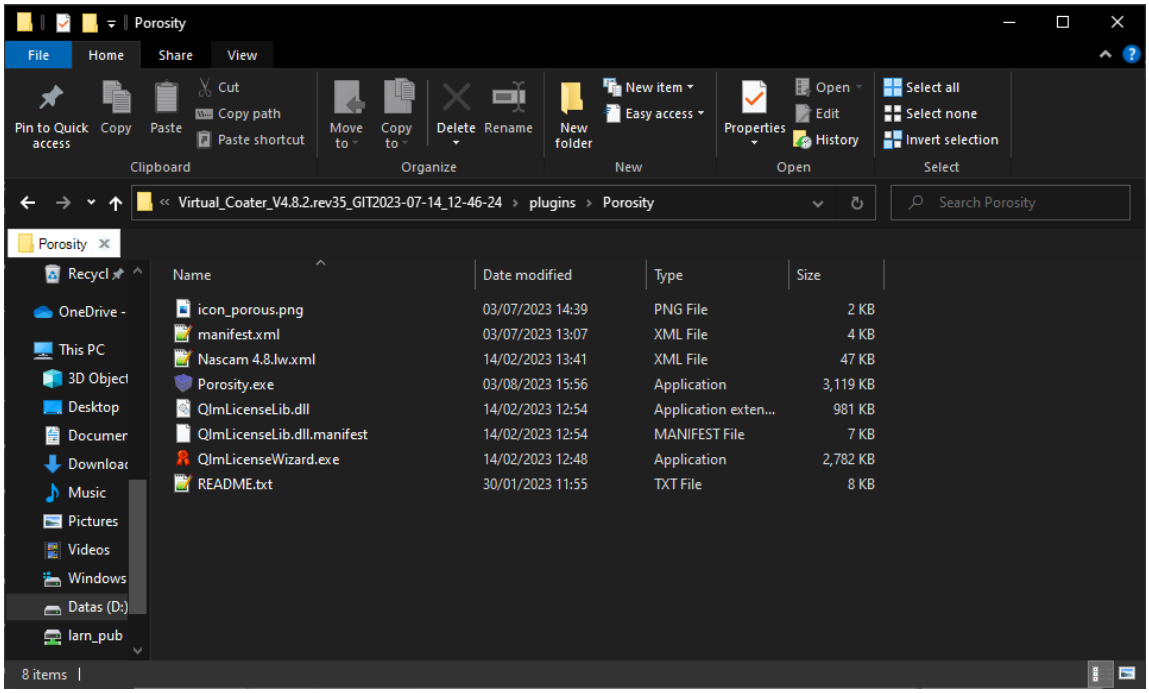
7.1.1 The plugin home folder

To be accessible from the **GUI** a Plugin needs to have a dedicated folder in the **plugins** folder of **Virtual Coater root directory**.

This folder should contain:

- **An executable:** *The plugin program*
- **An XML manifest:** *A document following certain rules that will tell the GUI info about the plugin in order to create a dedicated user interface frame.*
- **A readme text file:** *which will be accessible from the plugin frame.*

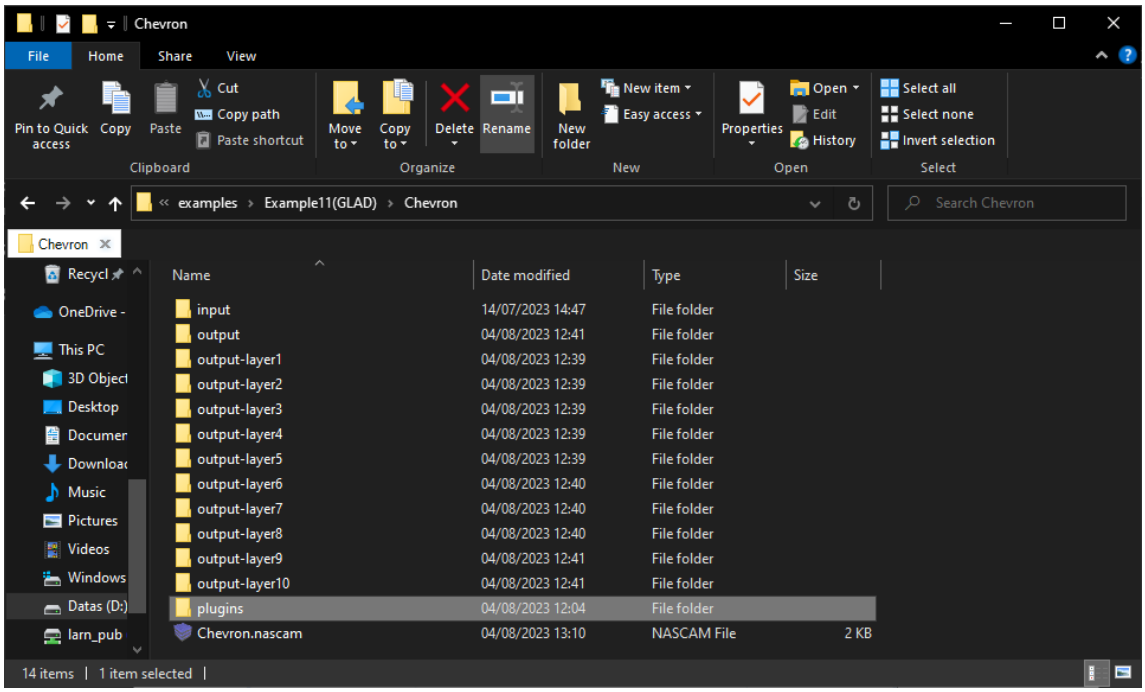




7.1.2 The plugin project folder

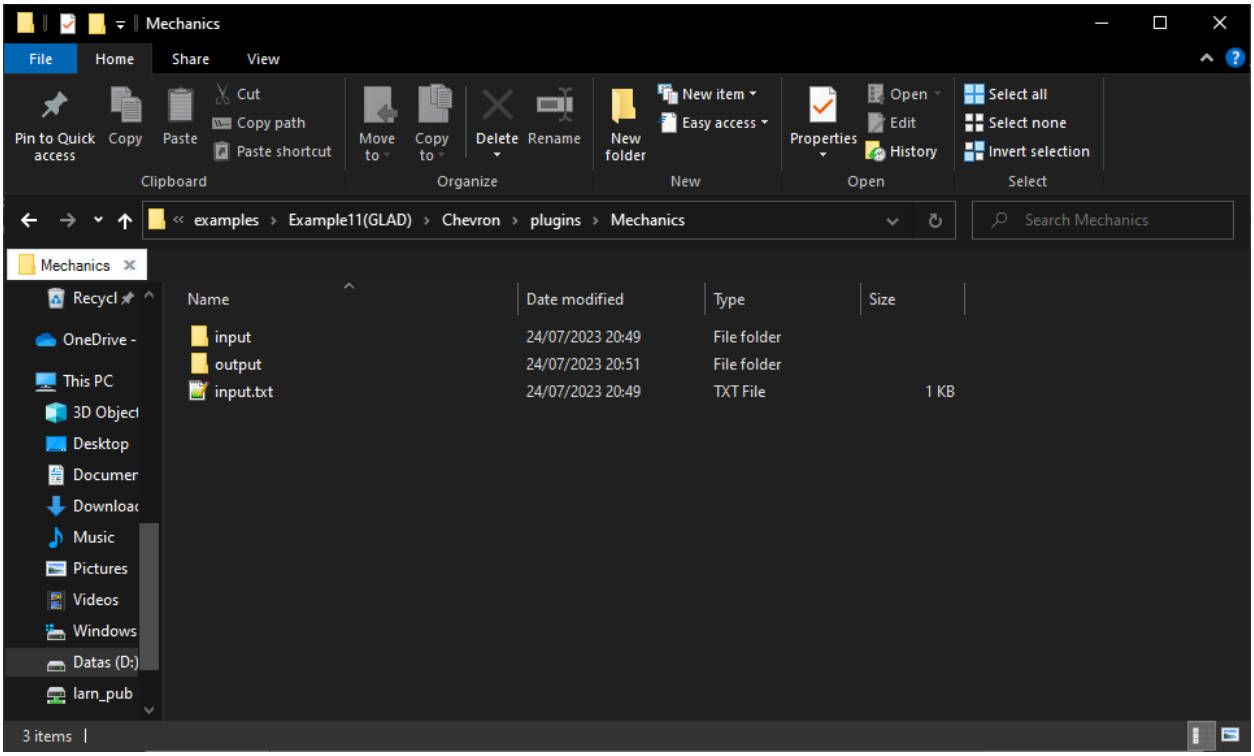
For each project a plugin folder is created if a plugin is opened. This folder will contain all the inputs and outputs of the plugin and the execution will be made at the root of the plugin folder inside the project folder.

The project folder:



Innovative Coating Solutions,
Place Saint-Pierre, 11, B-5380 Forville., Belgium
slu@incosol4u.com www.incosol4u.com

The plugin folder inside the project folder:



7.2 The manifest file

The `manifest file` is an **XML** (Extensible Markup Language) file. It is a language looking like **HTML** but anybody can create their own **tags**. The manifest file contains a **definite structure and a set of tags** that can be used to parse some important information about a plugin.

```
<plugin name="porosity">
    <info>
        <version>5.7</version>
        <executable>Porosity.exe</executable>
        <developer name="Jerome Muller" email="jerome.muller@unamur.be"/>
        <readme>README.txt</readme>
    </info>

    <requiredFiles>
        <file type="OUTPUT" >
            <name>coating.xyz</name>
        </file>
    </requiredFiles>
    <userInput>
        <group name="stack parameters">
            <param type="INT" >
                <name>number of studied layers</name>
                <default>1</default>
```

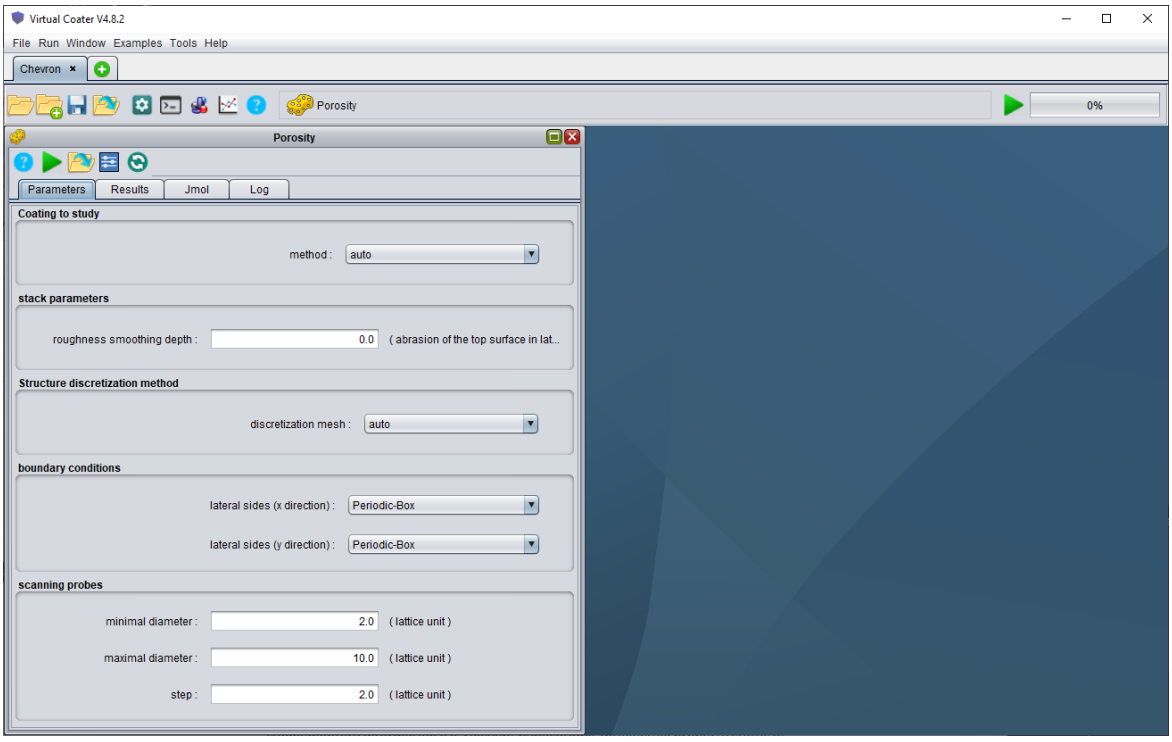
Innovative Coating Solutions,
Place Saint-Pierre, 11, B-5380 Forville., Belgium
slu@incosol4u.com www.incosol4u.com

```
            </param>
        </group>
        <group name="structure discretization method">
            <param type="LIST" ref="list_mesh">
                <name>discretization mesh</name>
                <element>cubic</element>
                <element>hexagonal</element>
                <element>manual</element>
            </param>
        </group>
        […]
```

```
    </userInput>
    <userOutput>
        <displayJmol>true</displayJmol>
        <displayGraph>true</displayGraph>
        <displayLog>true</displayLog>
    </userOutput>

</plugin>
```

This **manifest** will produce a plugin panel looking like:



Innovative Coating Solutions,
Place Saint-Pierre, 11, B-5380 Forville., Belgium
slu@incosol4u.com www.incosol4u.com

7.2.1 The structure of the manifest file

*All the fields marked with a * are **attributes**, meaning that you have to add them inside the corresponding opening tag*

- **info:**

```
<info>
    <version>5.7</version>
    <executable>Porosity.exe</executable>
    <icon>icon_porous.png</icon>
    <execdir>PROJECT</execdir>
    <command>$exe</command>
    <developer name="Jerome Muller" email="jerome.muller@unamur.be"/>
    <readme>README.txt</readme>
</info>
```

This is the info section and it contains all the general info about the plugin:

- **version:**
  The version number of this plugin.
- **executable:**
  The name of the executable.
- **execdir:**
  The directory where the plugin will be executed:
  - *ROOT: the plugin root folder*
  - *PROJECT: the plugin folder inside the project*
- **command:**
  The exact command to be run (will be executed from the plugin folder inside the project). This allows for having other kinds of executables (like python scripts, or java runnable jar etc…). In case of a .exe file this line is optional. Three special keywords are accessible to build the command as they will be replaced before execution:
  - *$project: the plugin folder (inside the project folder)*
  - *$root: the plugin root folder (with the manifest etc…)*
  - *$exe: the executable file absolute path*

Example:

With a java program:

```
<executable>my_plugin.jar</executable>
<command>java -jar -Xms1200m  my_plugin.jar</command>
```

Or with a python script:

Innovative Coating Solutions,
Place Saint-Pierre, 11, B-5380 Forville., Belgium
slu@incosol4u.com www.incosol4u.com

```
<executable>my_plugin.py</executable>
<command>python  my_plugin.py</command>
```

- **developer:**

  You can declare many developers and provide their name and email.

  - **name*:**

    Developer name.

  - **email:**

    Developer email.

- **readme:**

  The name of a readme file. It will be accessible from the GUI.

➢ **requiredFiles:**

```
<requiredFiles>
    <file type="OUTPUT" >
        <name>coating.xyz</name>
    </file>
    <file type="DATABASE" >
        <name>opticalDatabase.txt</name>
        <SQLquery>SELECT * FROM OpticalParameters</SQLquery>
    </file>
</requiredFiles>
```

The simulation input or output files needed. Plugins should not access files outside of the plugin folder inside each project, so in order to get the simulation results, the simulation parameters or some values from a database, plugin developers should declare here what are the files that their program needs. Those files can be of three types and each will require some different information:

- **file:**

  Each file bloc will declare a file needed for the execution of the plugin. In case of a multi-layer project the file of type *INPUT* or *OUTPUT* will be copied in separate folder for each layer. Whereas for *OUTPUT_TOTAL*, the final results of the simulation, they will be copied in the root of the input folder

  - **type*:**

    Possible types are: *INPUT, OUTPUT, DATABASE, EXTERNAL and OUTPUT_TOTAL*

  - **name:**

    The name of the file is required only for file types *OUTPUT, DATABASE and OUTPUT_TOTAL*.

Innovative Coating Solutions,
Place Saint-Pierre, 11, B-5380 Forville., Belgium
slu@incosol4u.com www.incosol4u.com

- **inputType:**

  This parameter is required only if the file type is set to *INPUT*. Must be one of: *SUBSTRATE, INPUT, ENERGY_METAL, ENERGY_GAS, ANGLE_METAL, ANGLE_GAS, and PATTERN.*

- **SQLquery:**

  This parameter is required only if the file type is set to ***DATABASE.*** It is a SQL command starting with select that will retrieve some data from the database. The result of the query to the database will be a file in the CSV format with semi column separators.

***Tip**: The name of the input file will be change to standard name ("input.txt", substrate.xyz", "pattern.txt", "energies_gas.txt", "angles_gas.txt", "energies_metal.txt", "angle_metal.txt"), even if the user set a different name to those file.*

userInput:

The *userInput* section list all the parameters that will be accessible to the final user via the GUI. They are grouped together for better readability, with the added possibility to show/hide groups.

- **group:**
  - **name*:**

    Name of this group of parameters, will be shown in the GUI as a titled panel.

  - **type*:**

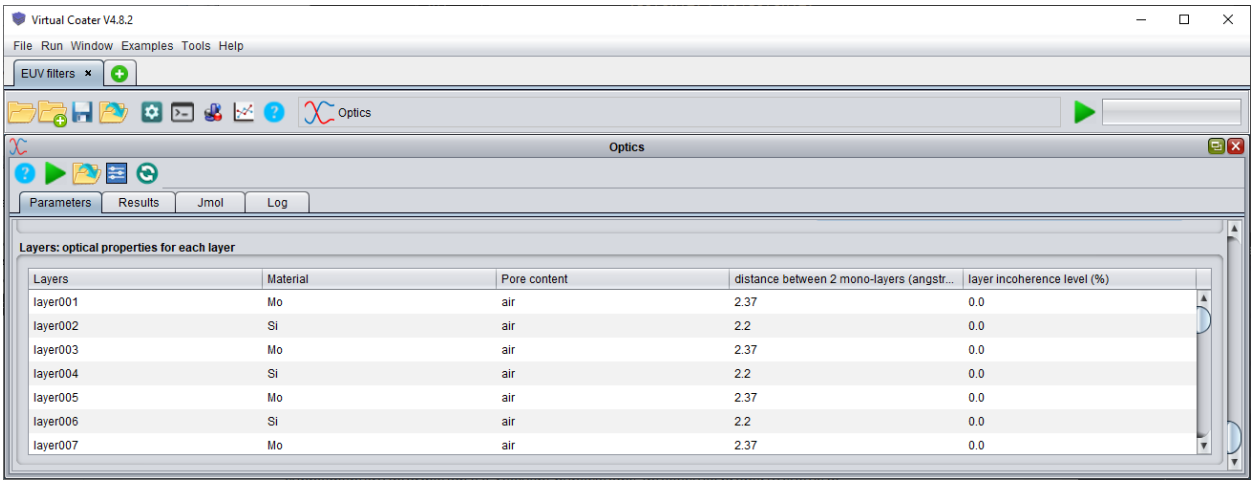
    Possible types are: *INPUT_TABLE, LAYER_TABLE* .Those are special groups that will display for the user a table where he will be able to select different value for respectively each layers or all layer associated to the different input files. This attribute is not required in other cases.

```
<group name="Optical Properties for layers" type="LAYER_TABLE">
    <param type="LISTDATABASE">
        <name>Material</name>
        <table>Data_Optic_Spectra</table>
        <column>Element</column>
        <column>Cu</column>
    </param>
    <param type="LISTDATABASE" >
        <name>Pore content</name>
        <table>Data_Optic_Spectra</table>
        <column>Element</column>
        <default>air</default>
```

Innovative Coating Solutions,
Place Saint-Pierre, 11, B-5380 Forville., Belgium
slu@incosol4u.com www.incosol4u.com

```
            </param>
            <param type="DOUBLE" >
                <name>distance between 2 mono-layers (angstroms)</name>
                <default>3.0</default>
            </param>
            <param type="DOUBLE" >
                <name>layer incoherence level (%)</name>
                <default>0.0</default>
            </param>
        </group>
```

These lines of code will produce in the GUI a table like that:



Layers	Material	Pore content	distance between 2 mono-layers (angstr...	layer incoherence level (%)
layer001	Mo	air	2.37	0.0
layer002	Si	air	2.2	0.0
layer003	Mo	air	2.37	0.0
layer004	Si	air	2.2	0.0
layer005	Mo	air	2.37	0.0
layer006	Si	air	2.2	0.0
layer007	Mo	air	2.37	0.0

When the plugin is executed the input.txt file will be like that:

Mo;air;2.37;0.0!Layers: optical properties for each layer kind-Layer 1
Si;air;2.2;0.0!Layers: optical properties for each layer kind-Layer 2
Mo;air;2.37;0.0!Layers: optical properties for each layer kind-Layer 3
Si;air;2.2;0.0!Layers: optical properties for each layer kind-Layer 4
Mo;air;2.37;0.0!Layers: optical properties for each layer kind-Layer 5
Si;air;2.2;0.0!Layers: optical properties for each layer kind-Layer 6
Mo;air;2.37;0.0!Layers: optical properties for each layer kind-Layer 7
[...]

- **displayIf*:**

  Used to show/hide some panel according to the value of some parameters.

```
    <group name="Additional Params" displayIf="list_mesh:cubic">
        <param type="DOUBLE" >
            <name>covalent radius (angstroms)</name>
```

```
            <default>1.0</default>
        </param>
    </group>
```

It requires that a parameter was attributed a reference first:

```
    <group name="structure discretization method">
        <param type="LIST" ref="list_mesh">
            <name>discretization mesh</name>
            <element>cubic</element>
            <element>hexagonal</element>
            <element>manual</element>
        </param>
    </group>
```

- **param:**

```
        <param type="DOUBLE" >
            <name>covalent radius</name>
            <default>1.0</default>
            <unit>Angstrom</unit>
        </param>
```

  - **type*:**

    Possible types are:

    DOUBLE: Double *[name, unit, default]*

    INT: Integer *[name, unit, default]*

    STRING: String *[name, default]*

    ATOM: List that contains all the first column of the periodic table from the database *[name, default]*

    FILE: File path text field with a file browser. The file selected by the user will be copied before the plugin execution *[name, default]*

    BOOLEAN: Checkbox that will be written in the input.txt file as "0" for unchecked and "1" for checked.

    LIST: List of element *[name, element (at least one)]*

    LISTDATABASE: List created from non-redundant element from the column of a database table *[name, table, column, default]*

    The names between square brackets are the associated fields

  - **name:** The name of the parameter
  - **unit:** The unit of the parameter

Innovative Coating Solutions,
Place Saint-Pierre, 11, B-5380 Forville., Belgium
slu@incosol4u.com www.incosol4u.com

- **default:** The default value of the parameter. User can always get back to default values with a button from the interface
- **element**: The lists elements
- **table**: The name of the database table associated with this *LISTDATABASE* parameter
- **column:** The name of the column of the database table associated with this *LISTDATABASE* parameter

- **userOutput:**

```
<userOutput>
    <displayJmol>true</displayJmol>
    <displayGraph>true</displayGraph>
    <displayLog>true</displayLog>
</userOutput>
```

Those Boolean fields allow you to show/hide some tab in the plugin interface. If they are absent the default is to show all tabs.

- **displayJmol:** Show/hide the structure viewer panel
- **displayGraph :** Show/hide the Graphics panel
- **displayLog:** Show/hide the log panel

7.2.2 Precision about the "requiredFiles" section

The **requiredFiles** section contains all the files that the plugin need for its execution it can be of three types:

- **OUTPUT** : This type of *requiredFiles* given a name will copy the associated input file from the project output folder, or if in multi-layer mode then it will copy this output file from each **output-layer[layerNb]** folder inside folders called **layer[layerNb]**

```
<file type="OUTPUT" >
    <name>coating.xyz</name>
</file>
```

- **INPUT**: This type of *requiredFiles* given a name will copy the associated input file from the project input folder. Same as for the *OUTPUT* file, they will be copied in folders named layer **layer[layerNb]**

```
<file type="INPUT">
    <inputType>SUBSTRATE</inputType>
</file>
```

Innovative Coating Solutions,
Place Saint-Pierre, 11, B-5380 Forville., Belgium
slu@incosol4u.com www.incosol4u.com

- **OUTPUT_TOTAL:** This type of *requiredFiles* will retrieve outputs of the simulation from the folder **output** of the project and copy them at the root of the **plugin input folder**.

```
<file type="OUTPUT_TOTAL" >
    <name>final_composition.csv</name>
</file>
```

- **DATABASE:** This type of *requiredFiles* will retrieve data from the database provided a "SQL query" and copy them at the root of the **plugin input folder**.

```
<file type="DATABASE" >
    <name>opticalDatabase.txt</name>
    <SQLquery>SELECT * FROM OpticalParameters</SQLquery>
</file>
```

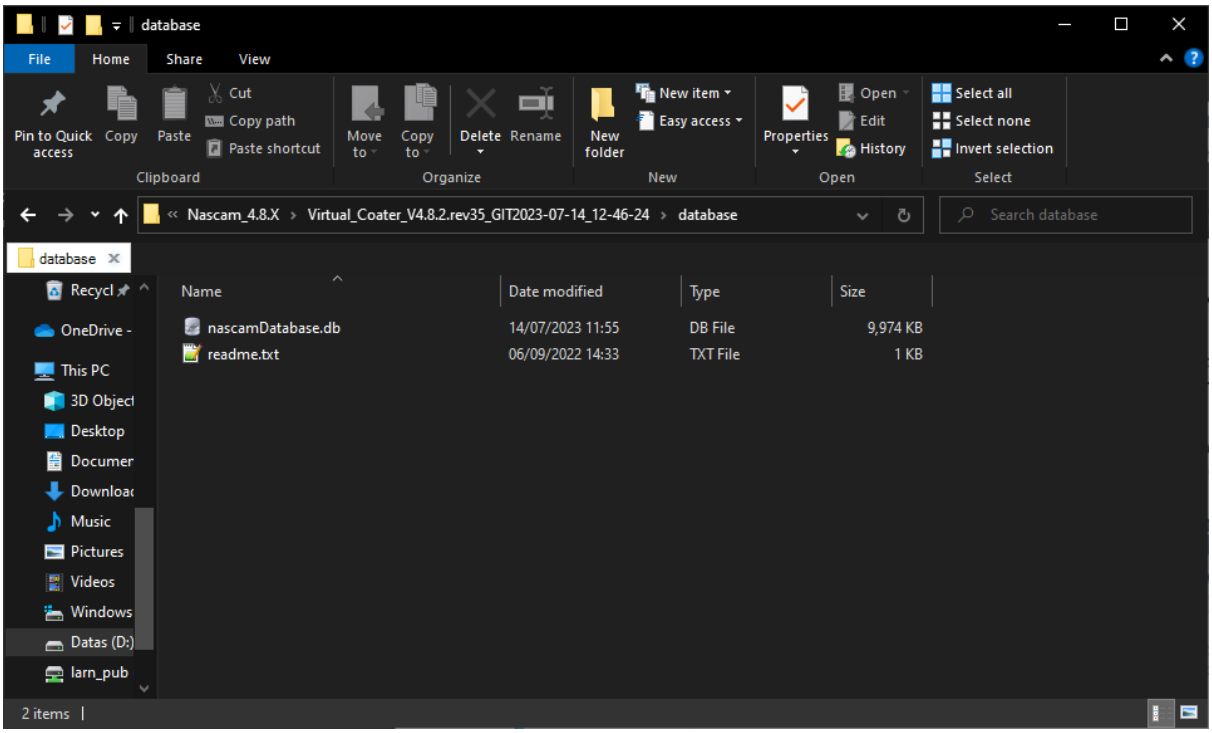
Innovative Coating Solutions,
Place Saint-Pierre, 11, B-5380 Forville., Belgium
slu@incosol4u.com www.incosol4u.com

8. The Database

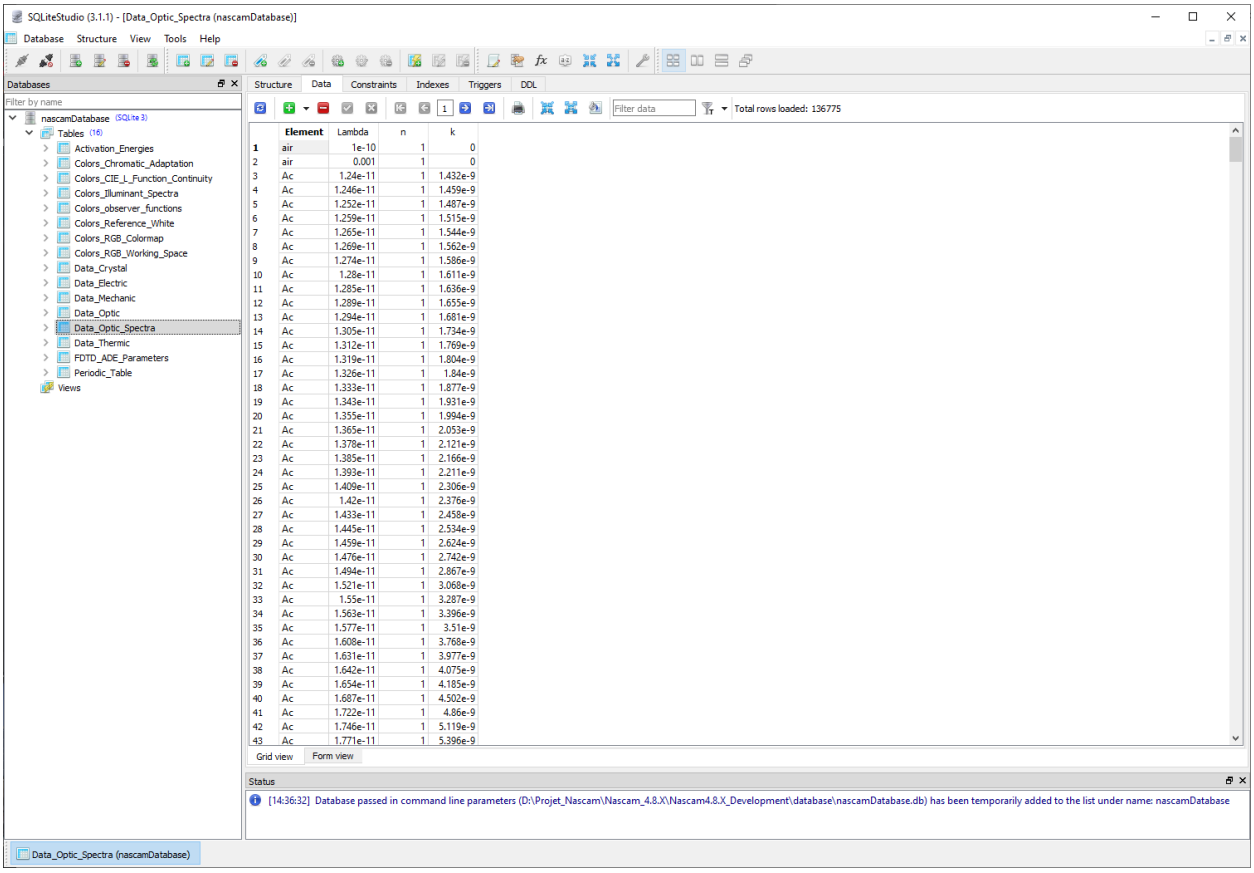
The Virtual Coater database purpose is to centralize all physical parameters in order to easily share and keep reference to them.

You can access and modify it through software like SQLiteStudio. You have also the option, for the Activation Energies table to modify it directly through Virtual Coater by creating new presets and summiting your changes.

If you would like to share your modification to the community, please send an email to stephane.lucas@unamur.be with your version of the database, and we will include it in subsequent version of Virtual Coater.



Innovative Coating Solutions,
Place Saint-Pierre, 11, B-5380 Forville., Belgium
slu@incosol4u.com www.incosol4u.com



Innovative Coating Solutions,
Place Saint-Pierre, 11, B-5380 Forville., Belgium
slu@incosol4u.com www.incosol4u.com

Appendix A: Summary of allowed fields for the manifest file

- Plugin
  - name*
  - info
    - version
    - executable
    - execdir
    - command
    - developer *(+)*
      - name*
      - email*
    - readme
  - requiredFiles
    - file *(+)*
      - type* [*OUTPUT_TOTAL, OUTPUT, INPUT, DATABASE, EXTERNAL*]
      - name
      - SQLquery
      - database
      - inputType [*SUBSTRATE,INPUT,ENERGIES,ANGLES,PATTERN*]
  - userInput
    - group *(+)*
      - name*
      - type* [*INPUT_TABLE,LAYER_TABLE*]
      - displayIf
      - param *(+)*
        - ref *
        - toPath*
        - name
        - type* [*DOUBLE,INT,STRING,ATOM,FILE,BOOLEAN,LIST,LISTDATABASE*]
        - unit
        - tooltip
        - default
        - write* [*ITEM, INDEX*]
        - table
        - column
  - userOutput
    - displayJmol
    - displayGraph
    - displayLog

Annotations : * *means that this field is an attribute,* (+) *means that this field can be repeated several times,* [A,B,C,…] *are the possible values of the corresponding field*