Introduction

SimLab Composer allows using Python, and Java Scripts to automate processes, both scripts automation is supported in two modes:

1. Command line (for batch processing for a large number of files). This includes command line without scripting using -ie command, and with scripting using Python, and Java scripts.
2. Interactive mode (from inside the GUI of SimLab Composer)

Scripting is supported in the Ultimate edition of SimLab Composer

Command line without scripting

Open the command line window, by typing "cmd" in Start. Go to the directory where SimLab Composer was installed, the default installation directory is "C:\Program Files\SimLab\SimLab Composer 10" to go there type **cd C:\Program Files\SimLab\SimLab Composer 10**

Now to run import/export functions in SimLab Composer, type Sim.. then start clicking the Tab button, until SimLabComposer.exe appears.

Type in the code -ie <import_file> <export_file>

With actual file locations, the below line will convert RubikCube.obj 3D models into RubikCube.skp in the indicated folders. Don't forget " "

-ie "C:\Users\simlab\Desktop\Delete\RubikCube.obj"
"C:\Users\simlab\Desktop\Delete\RubikCube.skp"

Check this article for more about the command line-based methods, also for commands on Mac.

Command line Python Scripts

Python scripts can be run from the command line using the following command

```
SimLabComposer.exe -py "File.py"
```

So if the user named a script as example.py, and saved it in folder C:\Scripts, The user should use the following command

```
SimLabComposer.exe -py "C:\Scripts\example.py"
```

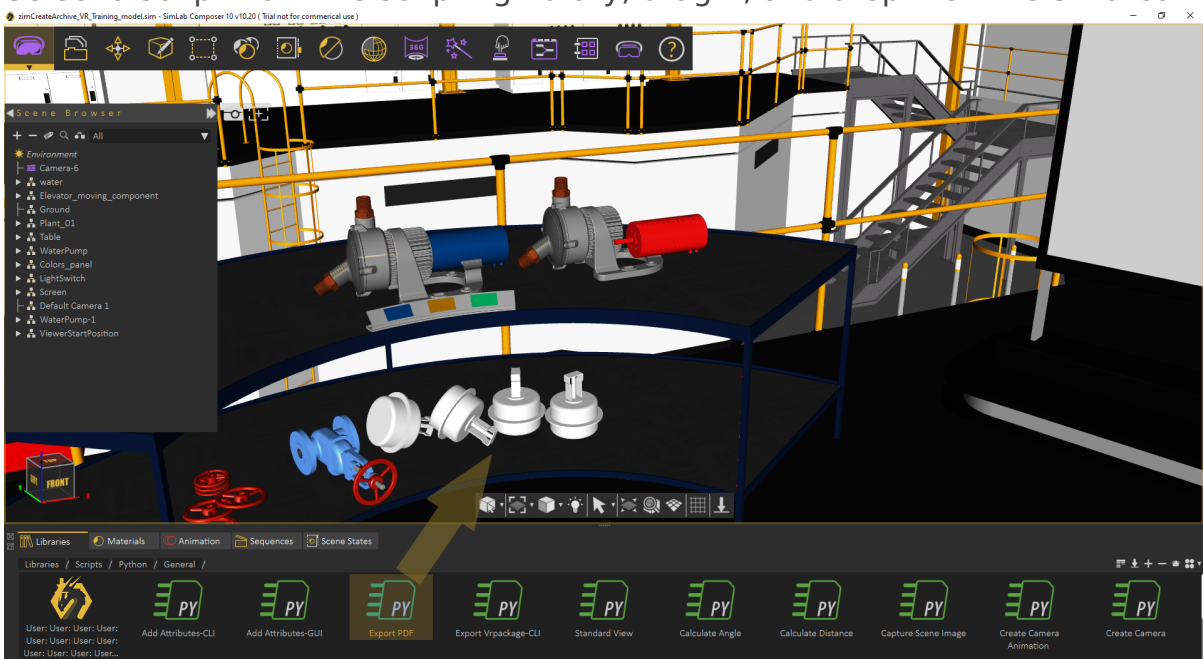# Passing arguments to Python Scripts

Passing arguments to a script makes it dynamic, and reusable without the need to change its code.

```
scene =Scene()

runtime =RunTime()

scene.reset()

fileName= runtime.args.getAsString("-path")

scene.importFile(fileName)
```
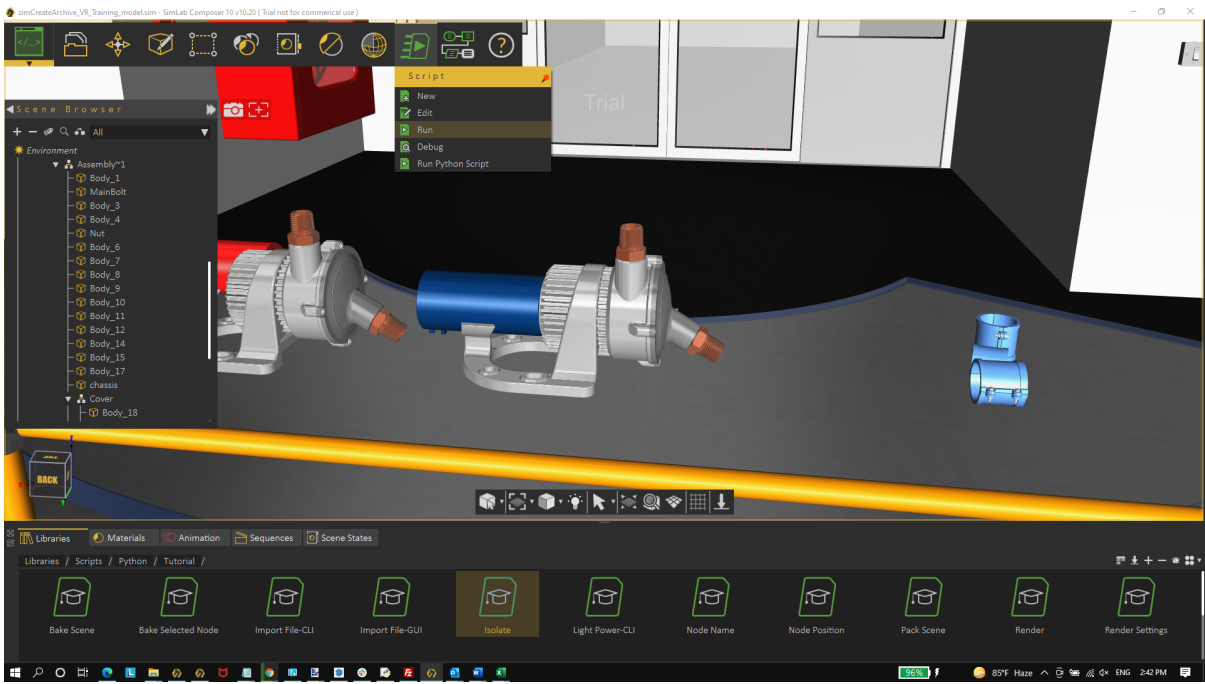
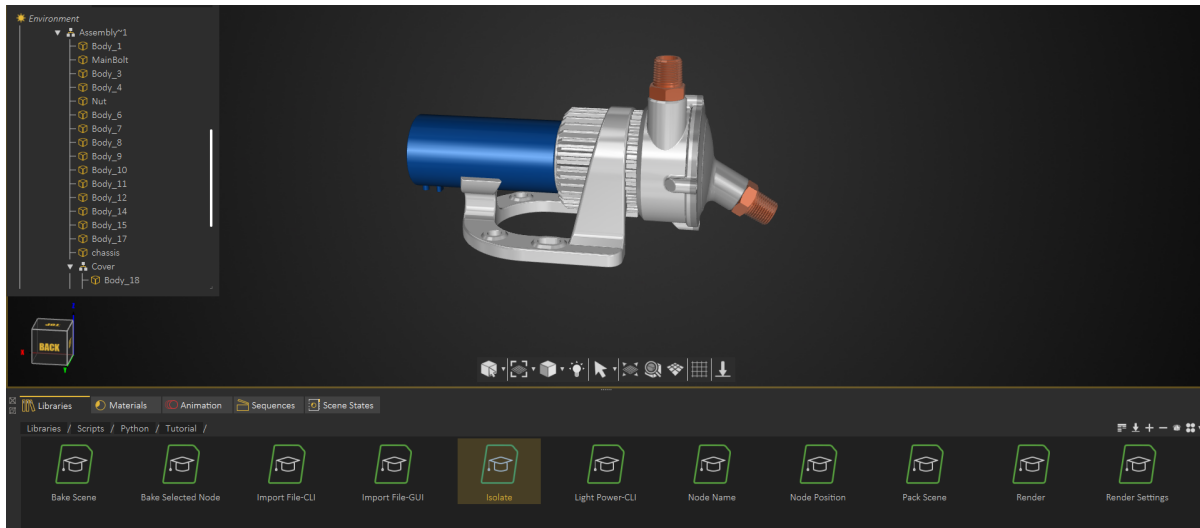# Interactive Scripting - Running Python script interactively

The user can run Python Scripts interactively in different ways:

1. Select a script from the scripting library, drag it, and drop it on the 3D area

2. Select a script from the library, then from the Script menu, click Run

3.

My first Python script / Python Scripts using GUI input

The following script gets the location to save the rendered image, using a GUI dialog. Renders the current scene, saves the resulting image in the selected location, and finally displays a dialog indicating that rendering is done.

```
from simlabpy import *


scene = Scene()

runtime = RunTime()

render_path = runtime.ui.getSaveFileName("Exported rendered image location:", "",

"*.jpg;;*.png")

scene.render(render_path)

runtime.ui.alert("Rendered image was created.")
```

For a list of supported Python scripting commands visit this page

Check out a blog about the approaches to automatically do things with SimLab Composer.

Revision #25
Created 9 September 2022 11:32:38 by Samia Sabri
Updated 27 December 2023 08:17:50 by Samia Sabri