Running VirtualLab Fusion Optical Simulations with Python

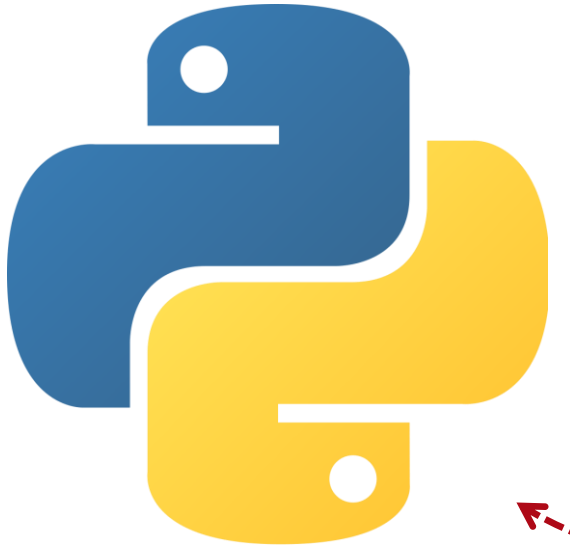
Abstract

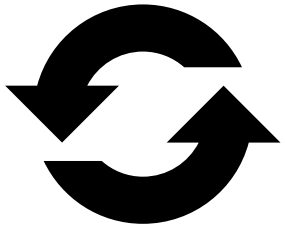
VirtualLab Fusion allows Python external access to its modeling technology, solvers and results. This use case is an introduction to a simple way of connecting Python to VirtualLab Fusion using the PATH-Variable and Visual Studio Code. In this example, we demonstrate how to run an optical simulation using a Python script to give the user a brief overview of this cross-platform simulation capability.

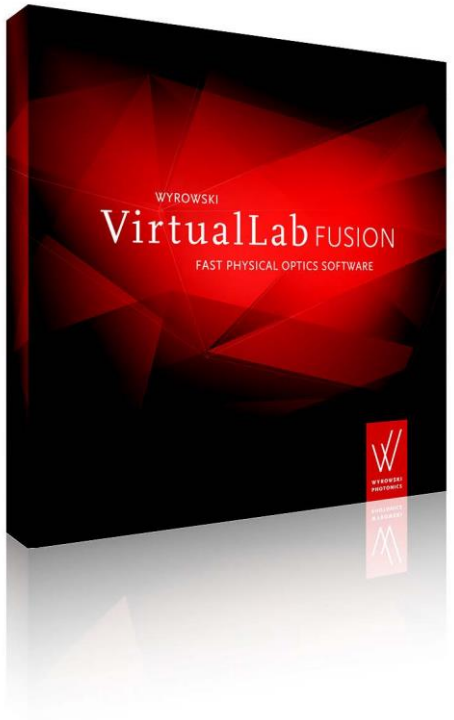
This Use Case Shows…

Python
- external functions

VirtualLab Fusion
- optical setup definition
- fast physical optics simulation engine



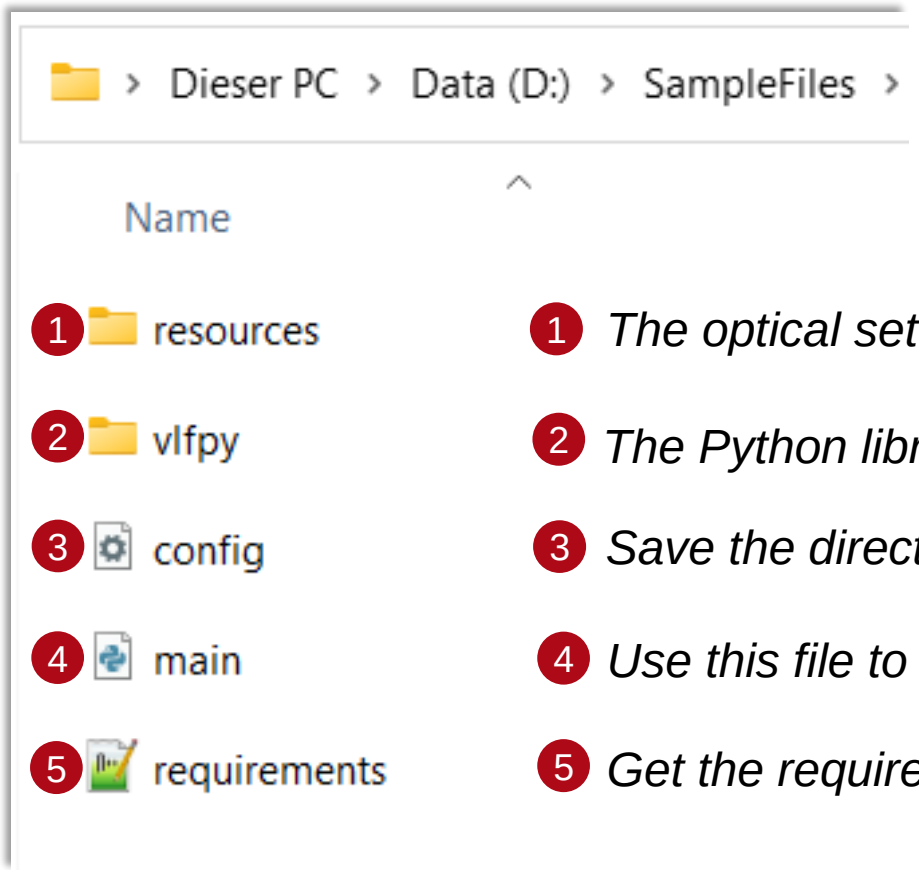




cross-platform simulation

Where to Find The Files

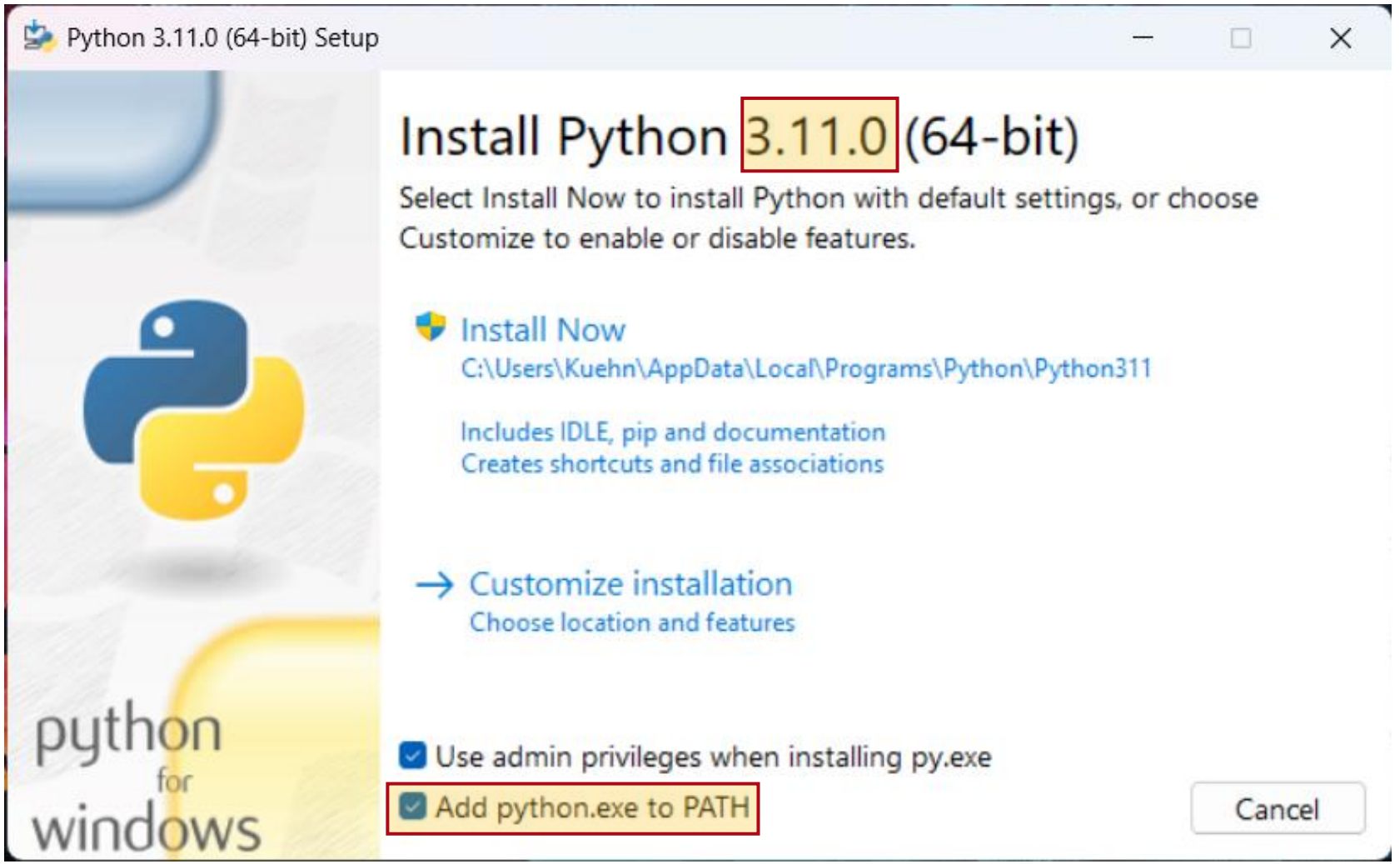
The user can find all files in the folder *SampleFiles*. The archive with the files can be downloaded from our website.



1. *The optical setup to be run is saved here*
2. *The Python library for simulation of optical setups*
3. *Save the directory of your VirtualLab Fusion here*
4. *Use this file to run your simulations*
5. *Get the required packages here*

Prepare Python

Make sure that **Python*** is installed on the computer. Notice that the option **Add python.exe to PATH** should be selected for installation. The instructions in this use case assume that no Python installation already exists on the computer.



* This use case has been created with Python 3.11.0.

Python Release Python 3.11.0 | Python.org

Prepare Python

Make sure that **Python 3.11.0** is installed on the computer. For demonstration purposes, we use the code editor Visual Studio Code (VS Code) as it offers a user-friendly installation workflow*. Of course, other Python editors can be used if desired.

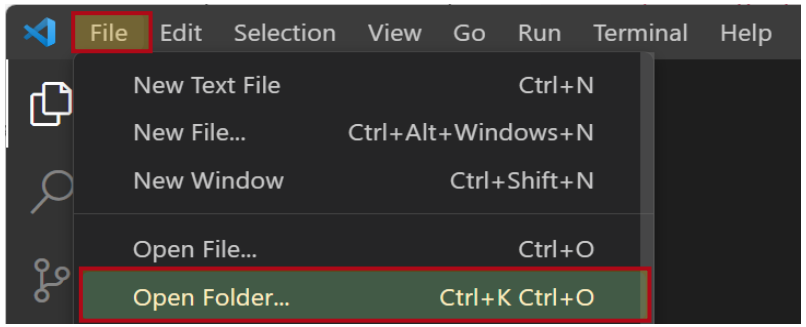
For the users who use VS Code:

1.1 Install the *Python Extension* from the *Visual Studio Marketplace*. The *Python Extension* is named "Python" and published by Microsoft.

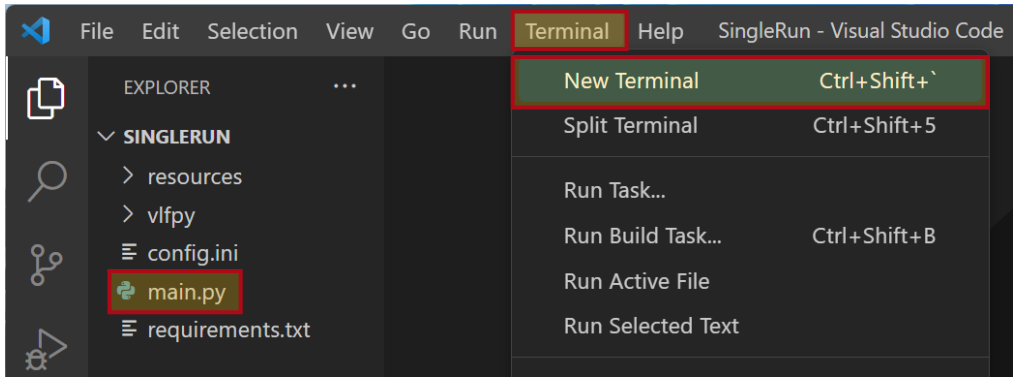
* For further information of the code editor Visual Studio Code for Python please read:
https://code.visualstudio.com/docs/python/python-tutorial
Note that here we demonstrate the installation of the required packages in the global environment. For users working with multiple Python projects, it is recommended to use project-specific virtual environments. Please also refer to the tutorial in the link above to create a virtual environment and install the required packages.

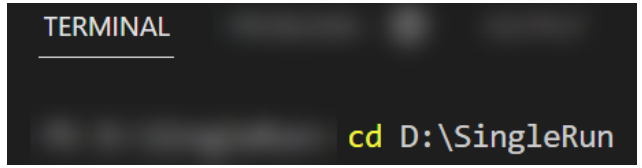
Prepare Python

1.2 Open the *SampleFiles* folder downloaded from our website with *File – Open Folder.*



1.3 Open a *Terminal* and change directory to the *SampleFiles* folder. Open the main.py file by clicking on it.

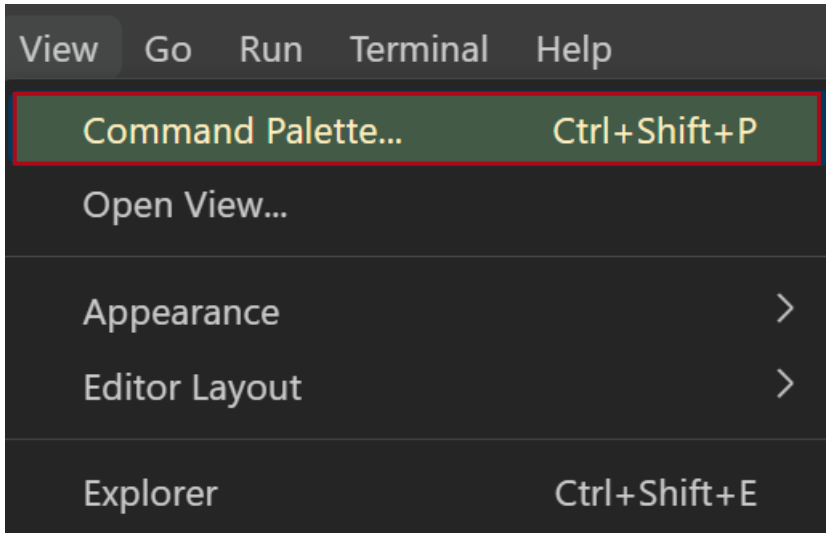


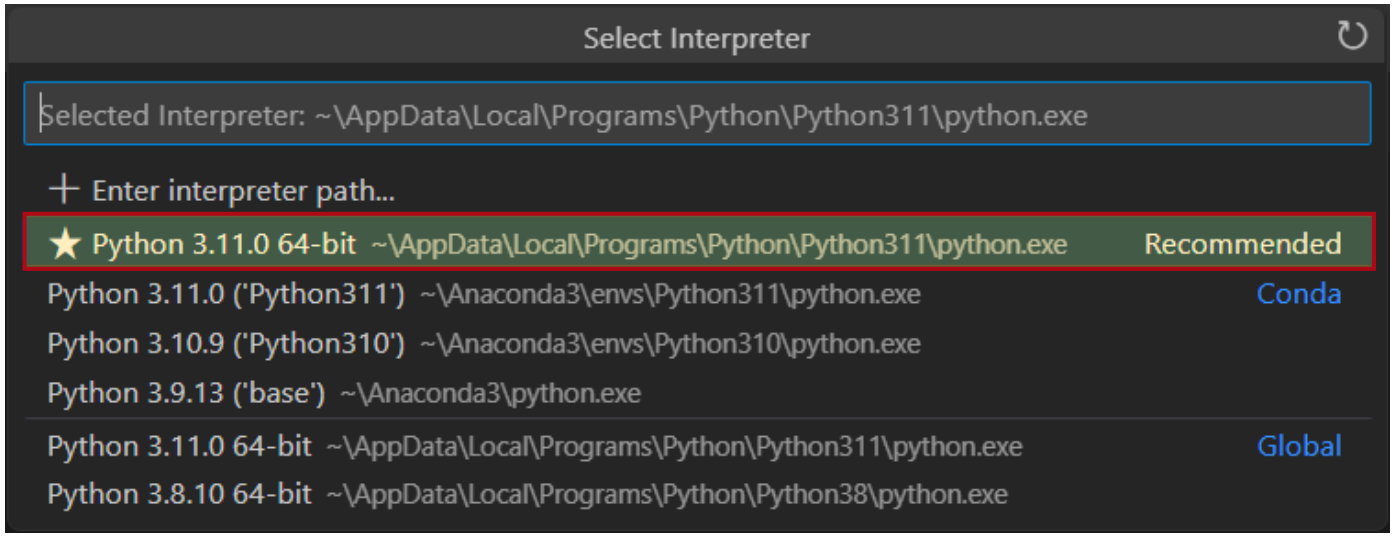


* For further information of the code editor Visual Studio Code for Python please read:
https://code.visualstudio.com/docs/python/python-tutorial
Note that here we demonstrate the installation of the required packages in the global environment. For users working with multiple Python projects, it is recommended to use project-specific virtual environments. Please also refer to the tutorial in the link above to create a virtual environment and install the required packages.

Prepare Python

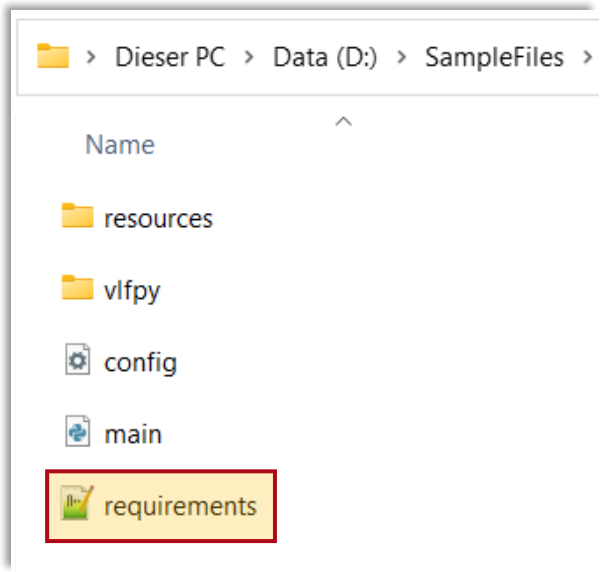
1.4 Open the *Command Palette* and type *Python: Select Interpreter,* make sure to choose **Python 3.11.0**. After this you can also see your choice in the status bar.





* For further information of the code editor Visual Studio Code for Python please read:
https://code.visualstudio.com/docs/python/python-tutorial
Note that here we demonstrate the installation of the required packages in the global environment. For users working with multiple Python projects, it is recommended to use project-specific virtual environments. Please also refer to the tutorial in the link above to create a virtual environment and install the required packages.

Prepare Python



1.5 The names of all required packages are saved in the file requirements.txt. Run the following command to make sure that all these packages are installed:
pip install -r requirements.txt

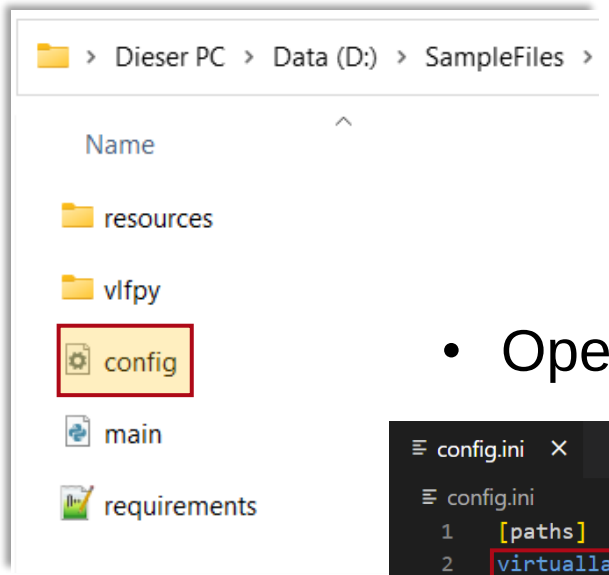
```
TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE

PS D:\SampleFiles> pip install -r requirements.txt
```

For users of other Python editors:

Please install all the packages saved in the file requirements.txt. One of the most common ways to do this is pip install: pip install -r requirements.txt

* For further information of the code editor Visual Studio Code for Python please read:
https://code.visualstudio.com/docs/python/python-tutorial
Note that here we demonstrate the installation of the required packages in the global environment. For users working with multiple Python projects, it is recommended to use project-specific virtual environments. Please also refer to the tutorial in the link above to create a virtual environment and install the required packages.

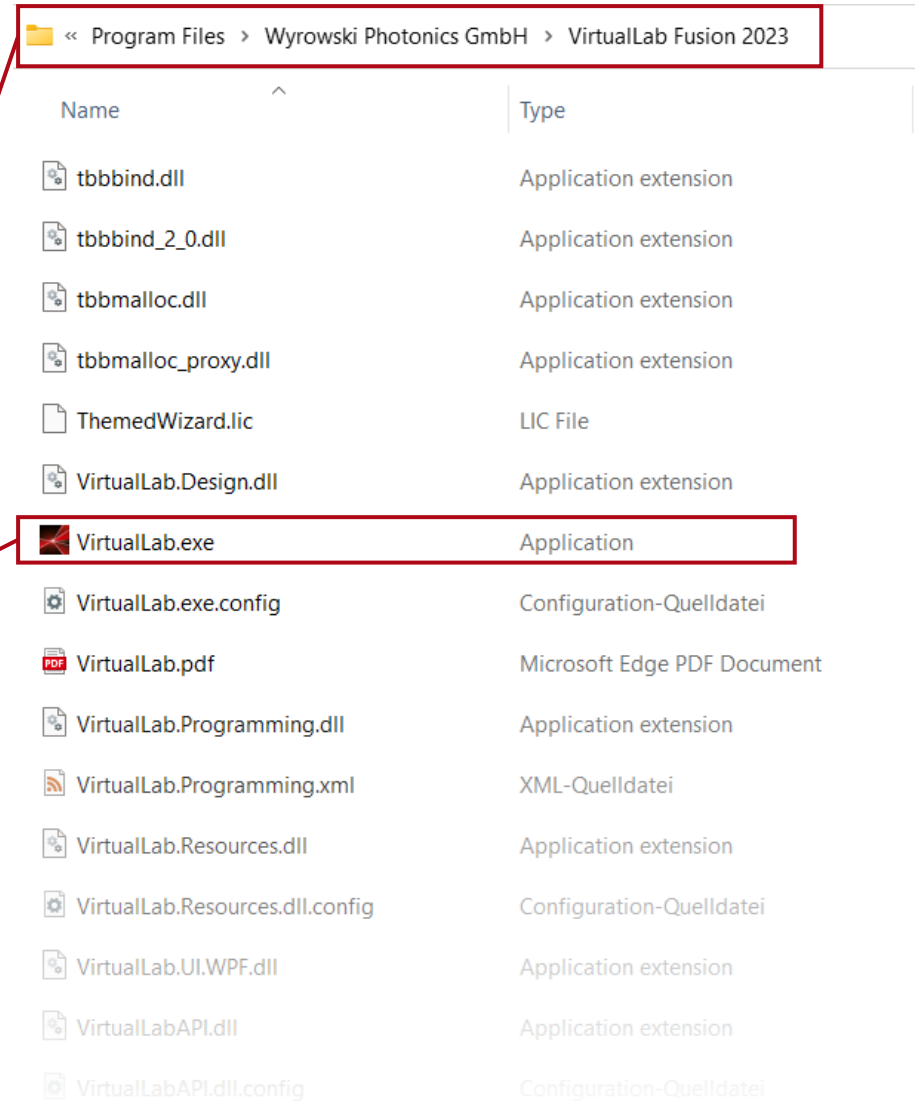
Configure the Path



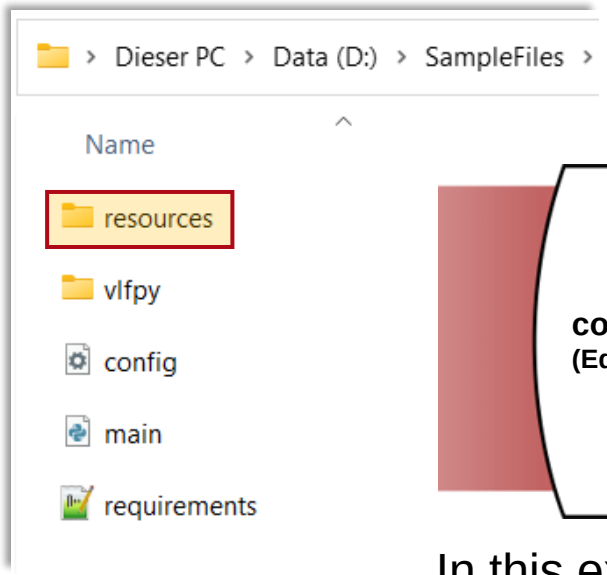
- Open the config.ini file.

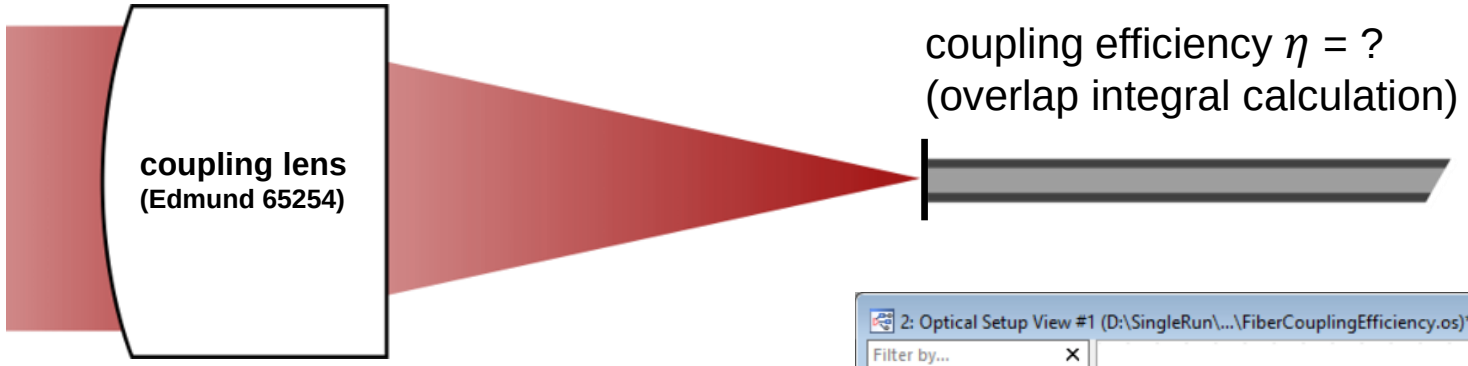
```
config.ini
1  [paths]
2  virtuallab = C:\Program Files\Wyrowski Photonics GmbH\VirtualLab Fusion 2023
3
4  [globals]
5  use_multicore = 1
6  number_of_cores = 12
```

- Set the directory of your VirtualLab Fusion installation (the folder which contains the VirtualLab.exe).



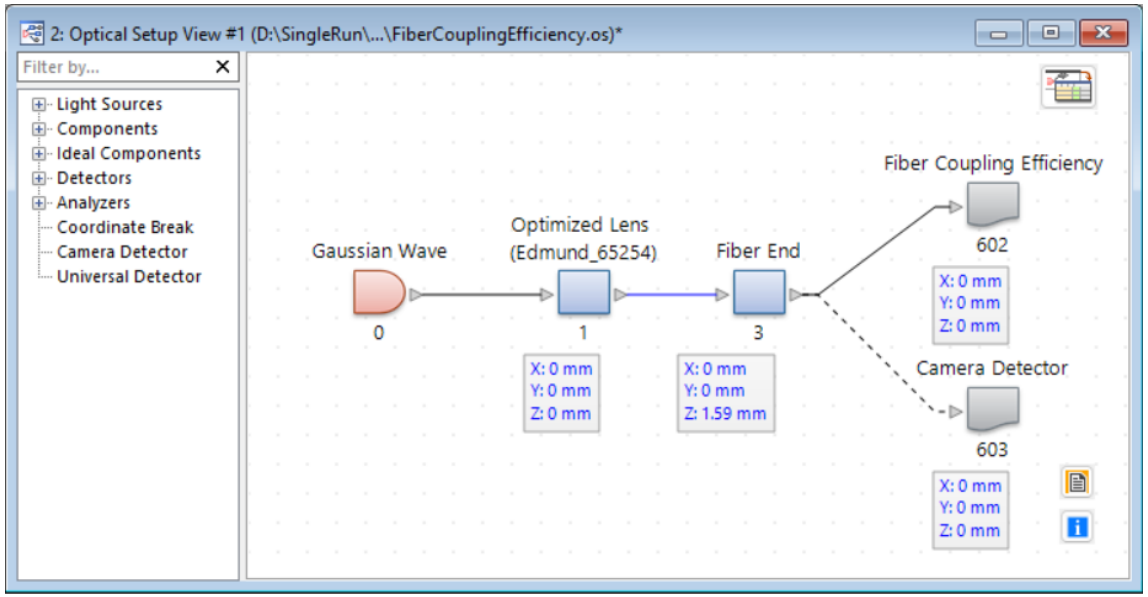
Define an Optical Setup in VirtualLab Fusion



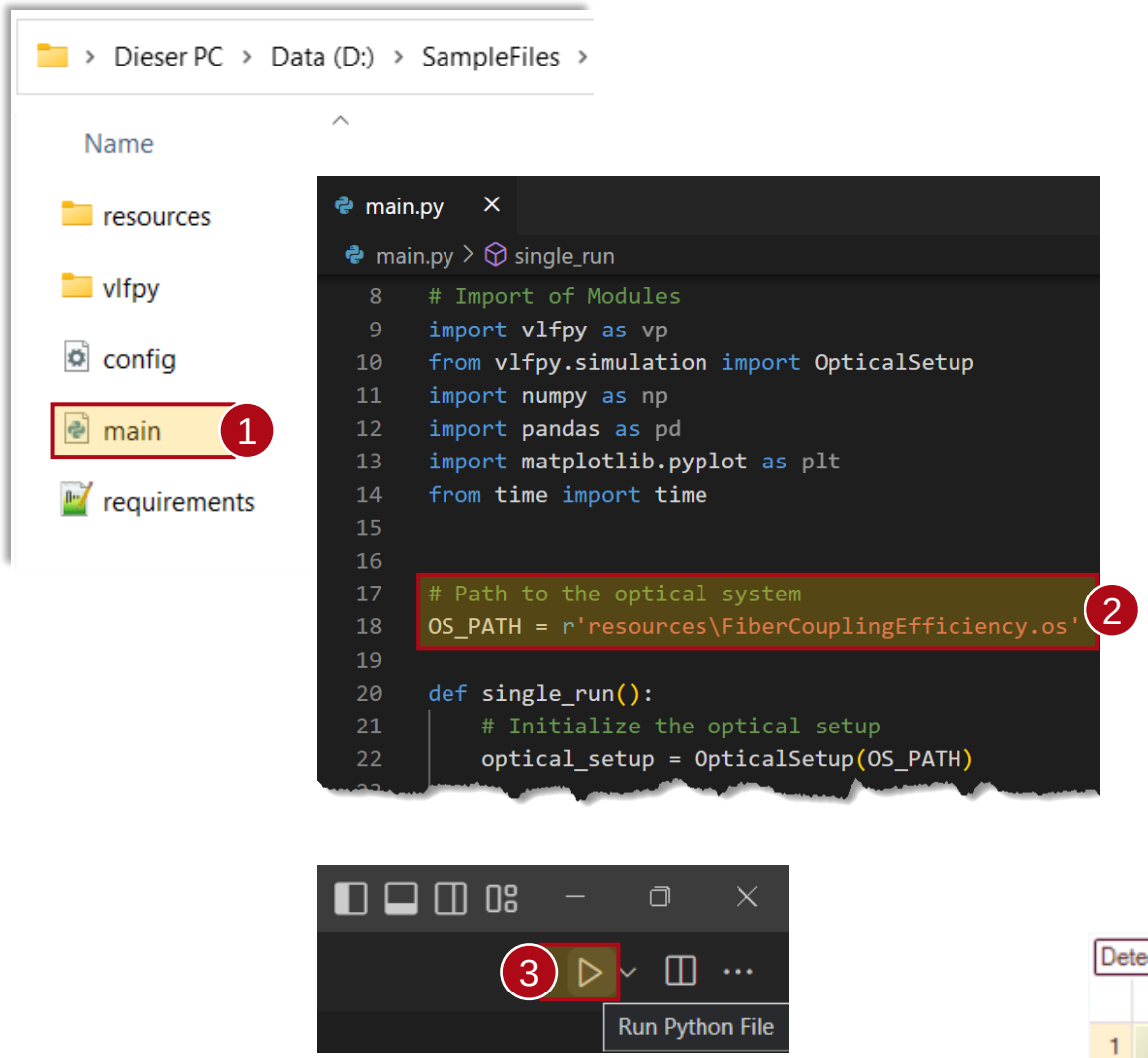


In this example we calculate the coupling efficiency of a fiber coupling lens. The corresponding optical setup in VirtualLab Fusion is defined and saved in the *resources* folder.

See the full example:
Parametric Optimization of Fiber-Coupling Lens



Run the Simulation



1.) Open the main.py file.

2.) Set the path to the optical setup to be evaluated. In this case, as mentioned in the previous page, the optical setup is saved in the *resources* folder.

3.) Press the play button at the upper right corner of the window to run the code.

In this example, the fiber coupling efficiency is displayed after executing the function.

```
TERMINAL

"Fiber Coupling Efficiency" (# 602)
        Singlemode Fiber Coupling Efficiency: 99.6 %
Simulation Time (s): 1.97
```

For comparison, this is the result if you calculate the coupling efficiency directly in VLF.

Detector Results

	Date/Time	Detector	Sub - Detector	Result
1		"Fiber Coupling Efficiency" (# 602) (Profile: General)	Singlemode Fiber Coupling Efficiency	99.6 %

Document Information

title	Execute an Optical Simulation in VirtualLab Fusion with Python
document code	CPF.0002
version	3.0
toolbox(es)	(depending on optical setup; for this example VirtualLab Fusion Basic)
• VLF version • Python version	• VirtualLab Fusion 2023.1 (Build 1.556) • Python 3.11.0
category	Feature Use Case
further reading	• Cross-Platform Optical Modeling and Design with VirtualLab Fusion and MATLAB • Parametric Optimization of Fiber-Coupling Lens • Cross-Platform Parameter Sweep with Python