CREATING STANDALONE EXECUTABLES – CONVERSION OF PY TO EXE

CONVERSION OF PYTHON TO EXECUTABLE (PY to EXE)

- By default, python script won't generate executable file.
- It is possible to convert python script to executable file in windows, linux, Mac, etc,.
- Python provides the following ways to convert py to exe file. They are
    1. Using **auto-py-to-exe**
    2. Using **pyinstaller** (pyinstaller package)
    3. Py2exe

1. USING AUTO-PY-TO-EXE INSTALLER

- This provides easy way to convert python script to standard executable file
- This GUI installer offers many options like input file selection, one directory or one file, console based or window based options, icon, etc,.

STEPS FOR INSTALLING AUTO-PY-TO-EXE

STEP 1:

- Run the pip command below to install auto-py-to-exe package installer.

```
pip install auto-py-to-exe
```

STEP 2:

- Run the command auto-py-to-exe to launch

```
auto-py-to-exe
```

Screenshot 1

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   JUPYTER                                  powershell

PS F:\VSC Projects Location\Python Lab\wxpython> pip install auto-py-to-exe
Collecting auto-py-to-exe
  Downloading auto_py_to_exe-2.23.1-py2.py3-none-any.whl (97 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 97.5/97.5 KB 507.6 kB/s eta 0:00:00
Collecting pyinstaller>=4.6
  Downloading pyinstaller-5.6-py3-none-win_amd64.whl (1.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.2/1.2 MB 1.0 MB/s eta 0:00:00
Collecting Eel==0.14.0
  Downloading Eel-0.14.0.tar.gz (17 kB)
  Preparing metadata (setup.py) ... done
Collecting bottle
  Downloading bottle-0.12.23-py3-none-any.whl (90 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 90.1/90.1 KB 639.1 kB/s eta 0:00:00
Collecting bottle-websocket
  Downloading bottle-websocket-0.2.9.tar.gz (2.0 kB)
  Preparing metadata (setup.py) ... done
Collecting future
  Downloading future-0.18.2.tar.gz (829 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 829.2/829.2 KB 409.5 kB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Collecting pyparsing
  Downloading pyparsing-3.0.9-py3-none-any.whl (98 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 98.3/98.3 KB 1.4 MB/s eta 0:00:00
Collecting whichcraft
  Downloading whichcraft-0.6.1-py2.py3-none-any.whl (5.2 kB)
Collecting pefile>=2022.5.30
  Downloading pefile-2022.5.30.tar.gz (72 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 72.9/72.9 KB 994.4 kB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: setuptools in c:\users\krishna\appdata\local\programs\python\python310\lib\site-packages (from pyinstall
er>=4.6->auto-py-to-exe) (58.1.0)
Collecting pywin32-ctypes>=0.2.0
```

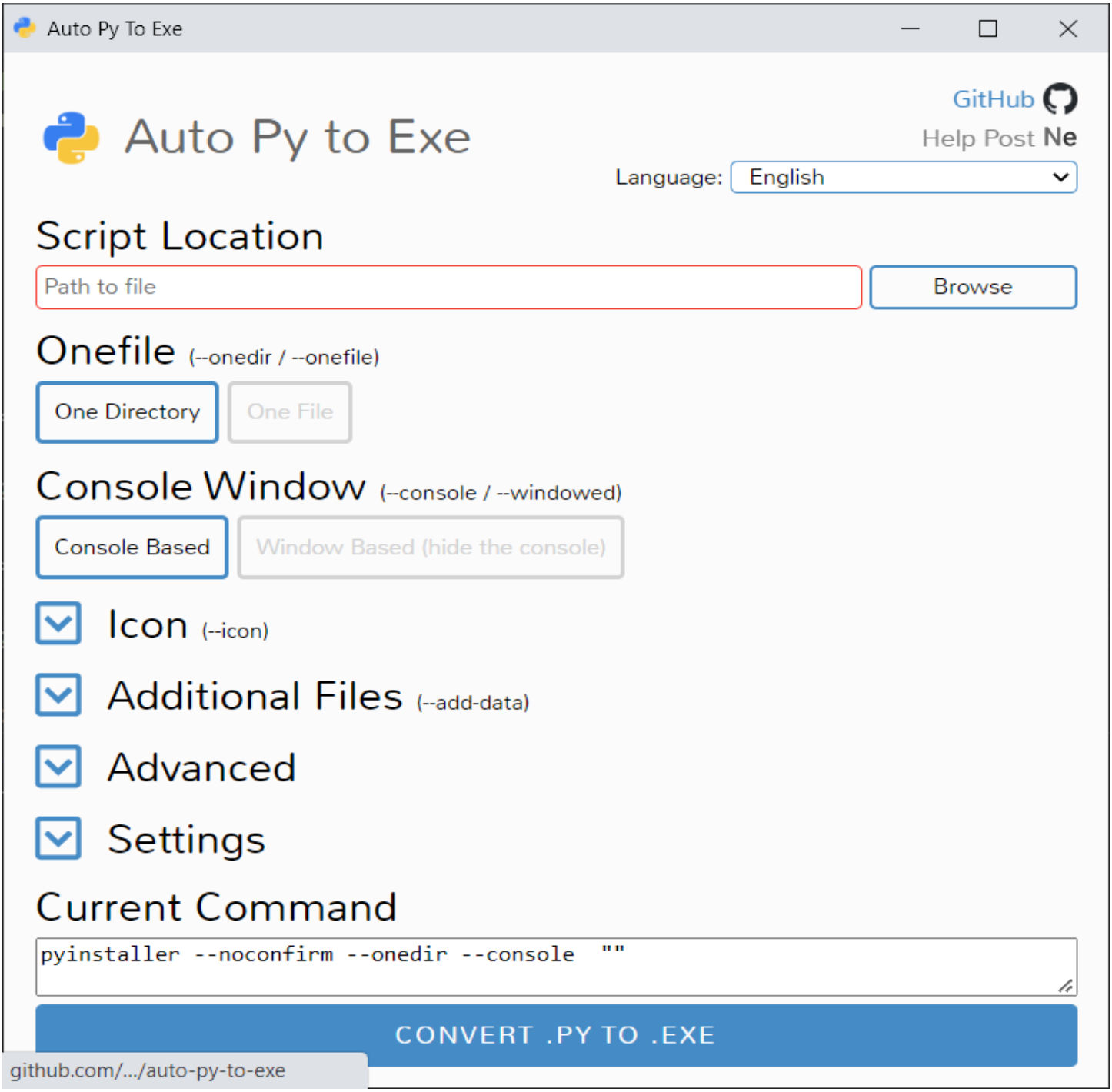
Screenshot 2

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   JUPYTER                                  powershell

  Downloading cffi-1.15.1-cp310-cp310-win_amd64.whl (179 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 179.1/179.1 KB 1.5 MB/s eta 0:00:00
Collecting pycparser
  Downloading pycparser-2.21-py2.py3-none-any.whl (118 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 118.7/118.7 KB 867.7 kB/s eta 0:00:00
Building wheels for collected packages: Eel, pefile, bottle-websocket, future
  Building wheel for Eel (setup.py) ... done
  Created wheel for Eel: filename=Eel-0.14.0-py3-none-any.whl size=17462 sha256=333563a0e2fd697422589ab0d43d0899756701b3e63e46d8691b344
81e29efe6
  Stored in directory: c:\users\krishna\appdata\local\pip\cache\wheels\5a\22\db\6e4ab5a6f105d4404fce7e17b453aa2ae920c70b04ac8ed294
  Building wheel for pefile (setup.py) ... done
  Created wheel for pefile: filename=pefile-2022.5.30-py3-none-any.whl size=69376 sha256=b9f497b5989b1bee577392a87e6ed7461ef5e5267c5dcb
0535d4ebc69bf25d20
  Stored in directory: c:\users\krishna\appdata\local\pip\cache\wheels\eb\60\37\ee40403cbd895ccdb57eb28b03b0afabeb449d5df9ce776a0d
  Building wheel for bottle-websocket (setup.py) ... done
  Created wheel for bottle-websocket: filename=bottle_websocket-0.2.9-py3-none-any.whl size=2348 sha256=2da14721181ad83dea785ae218f814c
b7582a8b29ce0c039b5d39c765d0e59e9
  Stored in directory: c:\users\krishna\appdata\local\pip\cache\wheels\d2\9c\7d\5cc2fe1ff85ad654a0e86d72d1706a97ef15db2200b83c98d6
  Building wheel for future (setup.py) ... done
  Created wheel for future: filename=future-0.18.2-py3-none-any.whl size=491070 sha256=b2f19a783e5b37f59368376cc7fae822cf08826708368930
e60dc28ce861ac21
  Stored in directory: c:\users\krishna\appdata\local\pip\cache\wheels\22\73\06\557dc4f4ef68179b9d763930d6eec26b88ed7c389b19588a1c
Successfully built Eel pefile bottle-websocket future
Installing collected packages: whichcraft, pywin32-ctypes, bottle, altgraph, zope.interface, zope.event, pyparsing, pyinstaller-hooks-c
ontrib, pycparser, greenlet, future, pefile, cffi, pyinstaller, gevent, gevent-websocket, bottle-websocket, Eel, auto-py-to-exe
Successfully installed Eel-0.14.0 altgraph-0.17.3 auto-py-to-exe-2.23.1 bottle-0.12.23 bottle-websocket-0.2.9 cffi-1.15.1 future-0.18.2
 gevent-22.10.1 gevent-websocket-0.10.1 greenlet-1.1.3.post0 pefile-2022.5.30 pycparser-2.21 pyinstaller-5.6 pyinstaller-hooks-contrib-
2022.10 pyparsing-3.0.9 pywin32-ctypes-0.2.0 whichcraft-0.6.1 zope.event-4.5.0 zope.interface-5.5.0
WARNING: You are using pip version 22.0.4; however, version 22.3 is available.
You should consider upgrading via the 'C:\Users\Krishna\AppData\Local\Programs\Python\Python310\python.exe -m pip install --upgrade pip
' command.
PS F:\VSC Projects Location\Python Lab\wxpython>
```

Screenshot 3

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    JUPYTER

PS F:\VSC Projects Location\Python Lab\wxpython> auto-py-to-exe
```

HOME PAGE OF AUTO PY TO EXE



INPUT OPTIONS IN AUTO PY TO EXE

Script location

- Here input python file must be given using browse button.

One File

- This option will **create a single executable file which contains all dependencies** but not media files.

One Directory

- This option will **create an executable and place all the dependencies in a single location** (single directory)

Additional files

- Here you can add any additional files if necessary
- The selected files from the “Additional Files” option will not be included in the .exe file if the option “One File” is selected.

I. CONVERSION OF PY TO EXE USING AUTO PY TO EXE

Language	:	Python 3
Editor	:	VSC Editor
OS	:	Windows 10
GUI Framework	:	wxPython
Installer	:	**auto-py-to-exe**

1. SOURCE CODE

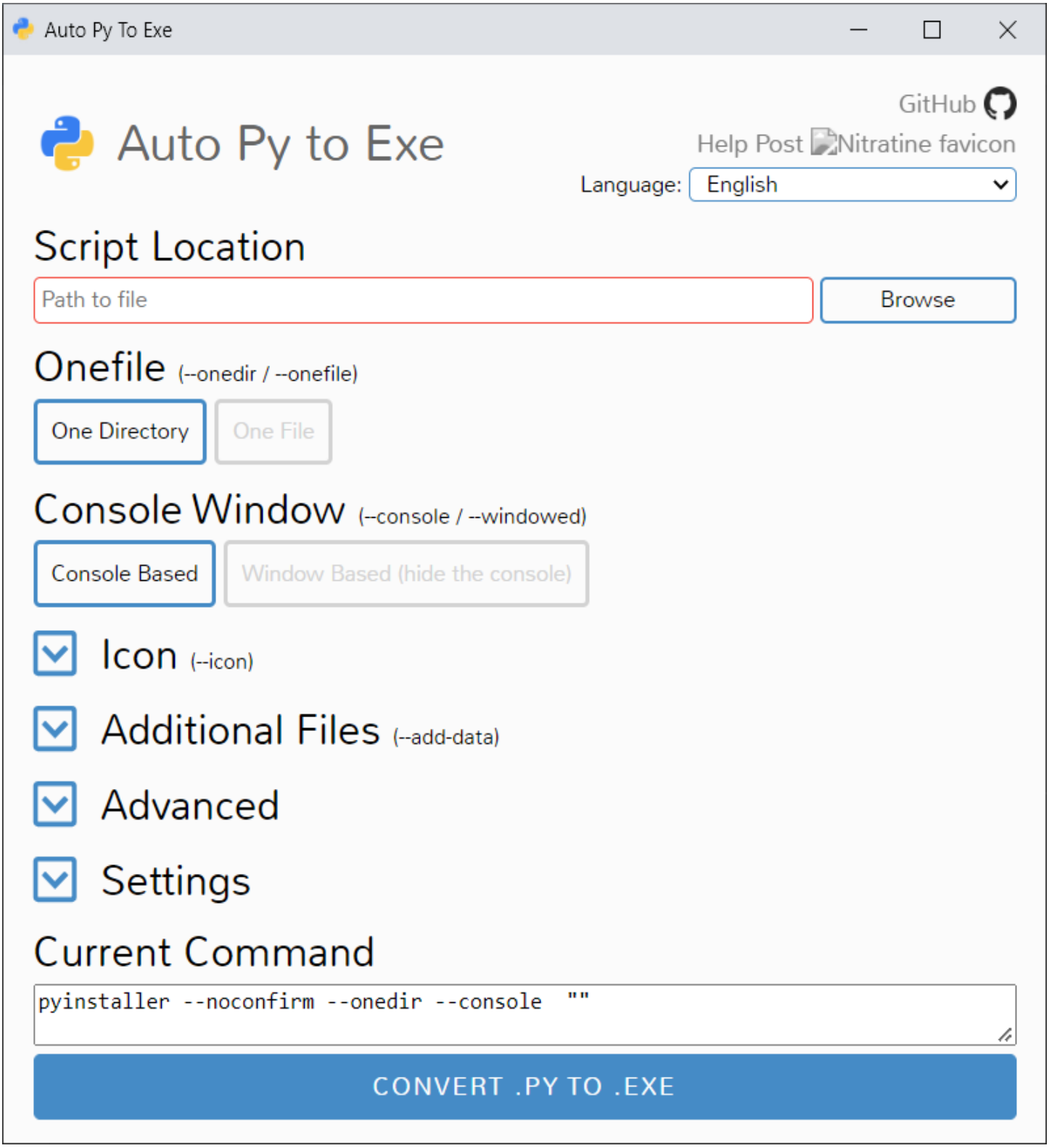
```
import wx
# create an object for application class
ob=wx.App()
# create a root window
rt=wx.Frame(None, title="Addition", size=(420,490))
# create a panel layout
pl=wx.Panel(rt)
# BUTTON EVENT HANDLER 1
def disp(et):
# get the first input from the user via first text box
    u1=tb1.GetValue()
# convert string formatted number to number
    a=int(u1)
# get the second input from the user via second text box
    u2=tb2.GetValue()
# convert string to integer
    b=int(u2)
# add two numbers
    c=a+b
```

```
# display the result in mult-line text box which accepts only string data
    rs.SetValue("Sum is: "+str(c))
# BUTTON EVENT HANDLER 2
def clearUI(et):
    tb1.SetValue("")
    tb2.SetValue("")
    rs.SetValue("")
# BUTTON EVENT HANDLER 3
def skipApp(et):
    wx.Exit()
# add label 1
l1=wx.StaticText(pl, label="Number 1: ", pos=(15,15))
# add text box 1
tb1=wx.TextCtrl(pl, size=(290,25), pos=(15,40))
# add label 2
l2=wx.StaticText(pl, label="Number 2: ", pos=(15,75))
# add text box 2
tb2=wx.TextCtrl(pl, size=(290,25), pos=(15,100))
# add button 1, 2, 3
b1=wx.Button(pl, label="Add", pos=(15,135))
# add event handler to button 1
b1.Bind(wx.EVT_BUTTON,disp)
b2=wx.Button(pl, label="Clear", pos=(115,135))
# add event handler to button 2
b2.Bind(wx.EVT_BUTTON, clearUI)
# add event handler to button 3
b3=wx.Button(pl, label="Exit", pos=(215,135))
b3.Bind(wx.EVT_BUTTON, skipApp)
```

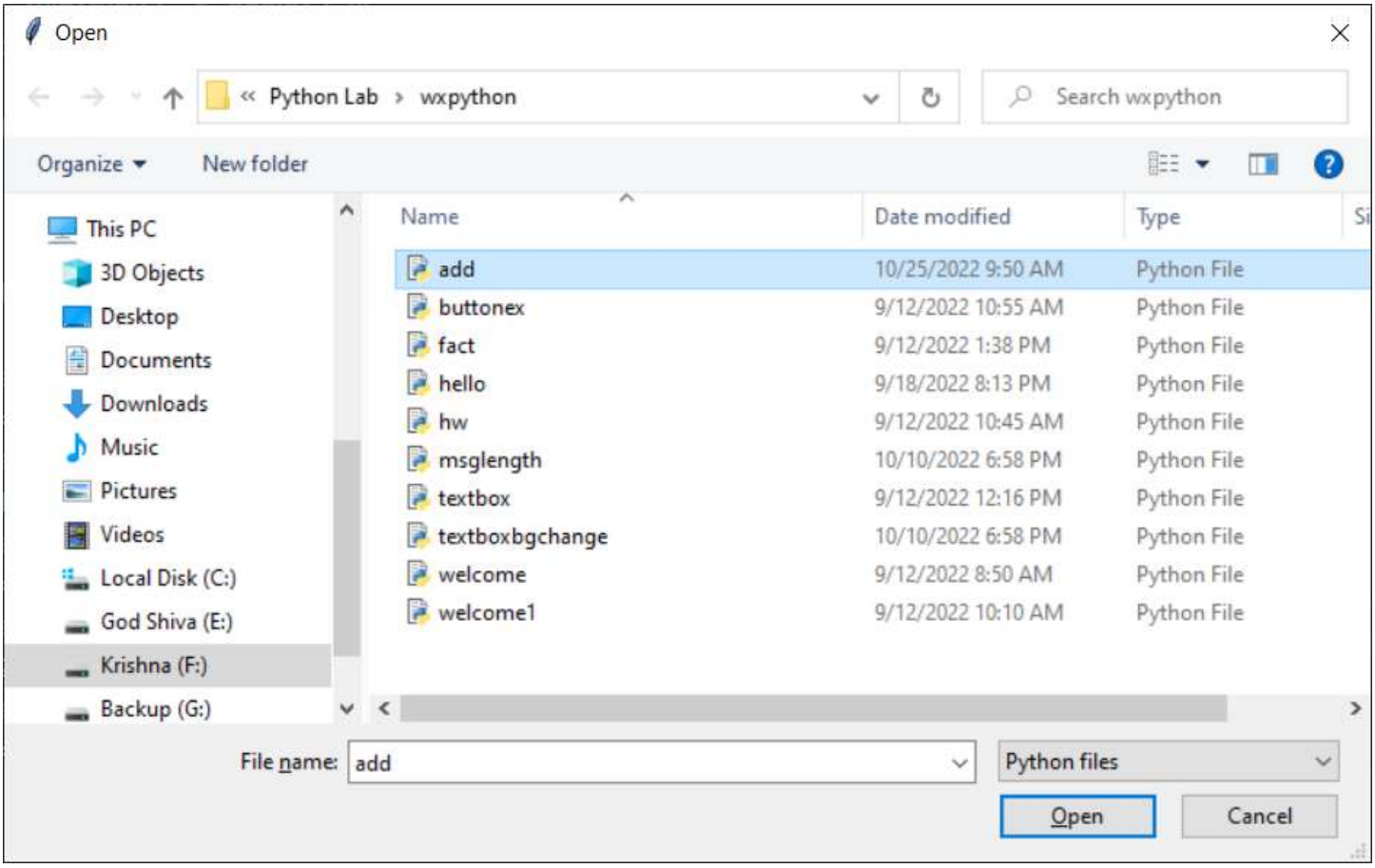
```
# add label
lb=wx.StaticText(pl, label="Result: ", pos=(15, 170))
# add multi-line text box
rs=wx.TextCtrl(pl, size=(290,80), pos=(15,195))
# activate and display root window
rt.Show()
# show the window in center screen
rt.Centre()
# run the application
ob.MainLoop()
```

2. OUTPUT

2.1 STARTING AUTO-PY-TO-EXE INSTALLER



2.2 INPUT FILE SELECTION

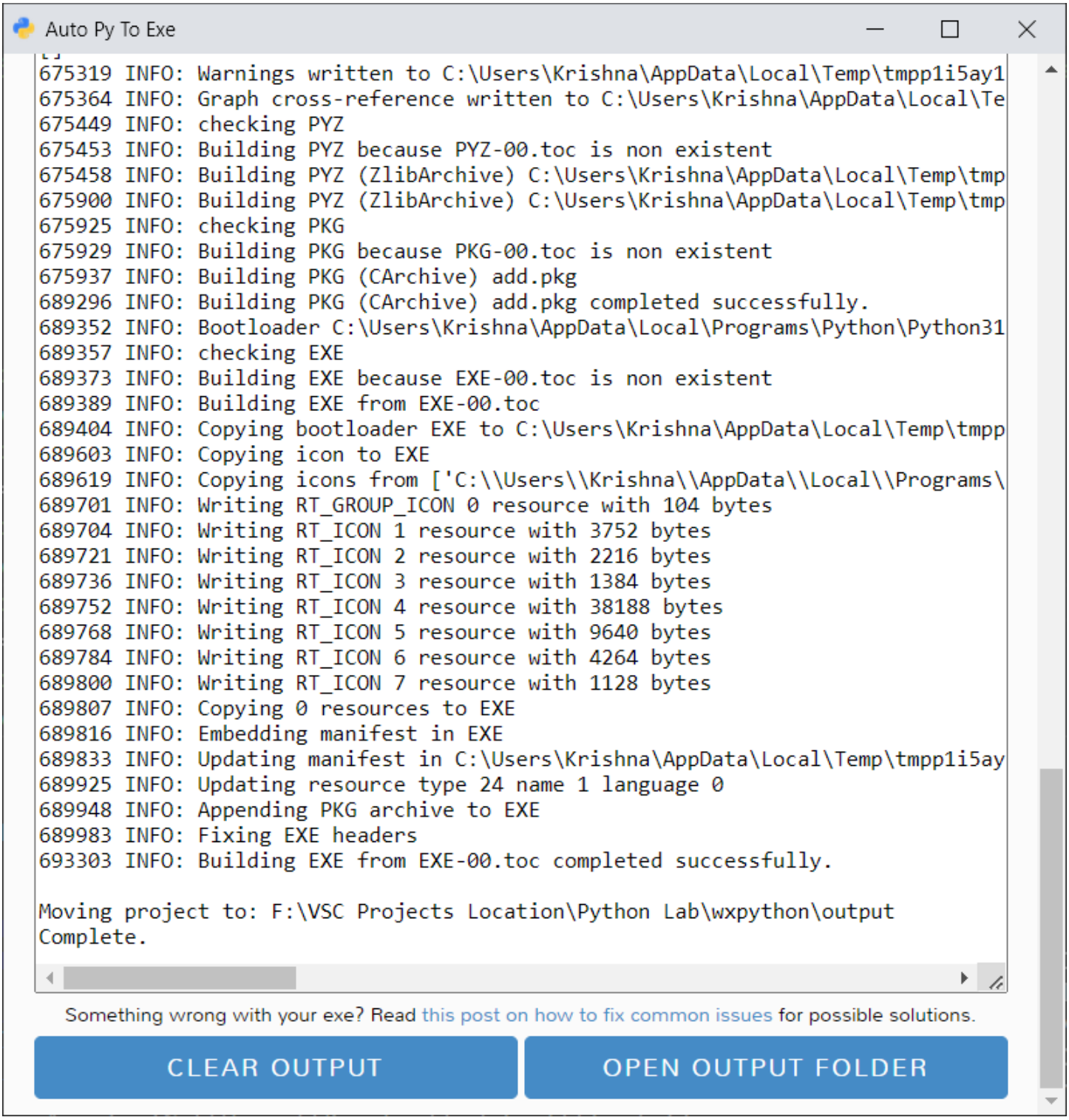


2.3 SELECT THE REQUIRED OPTIONS

1. Select the one file
2. Select the Window Based (hide the console window)

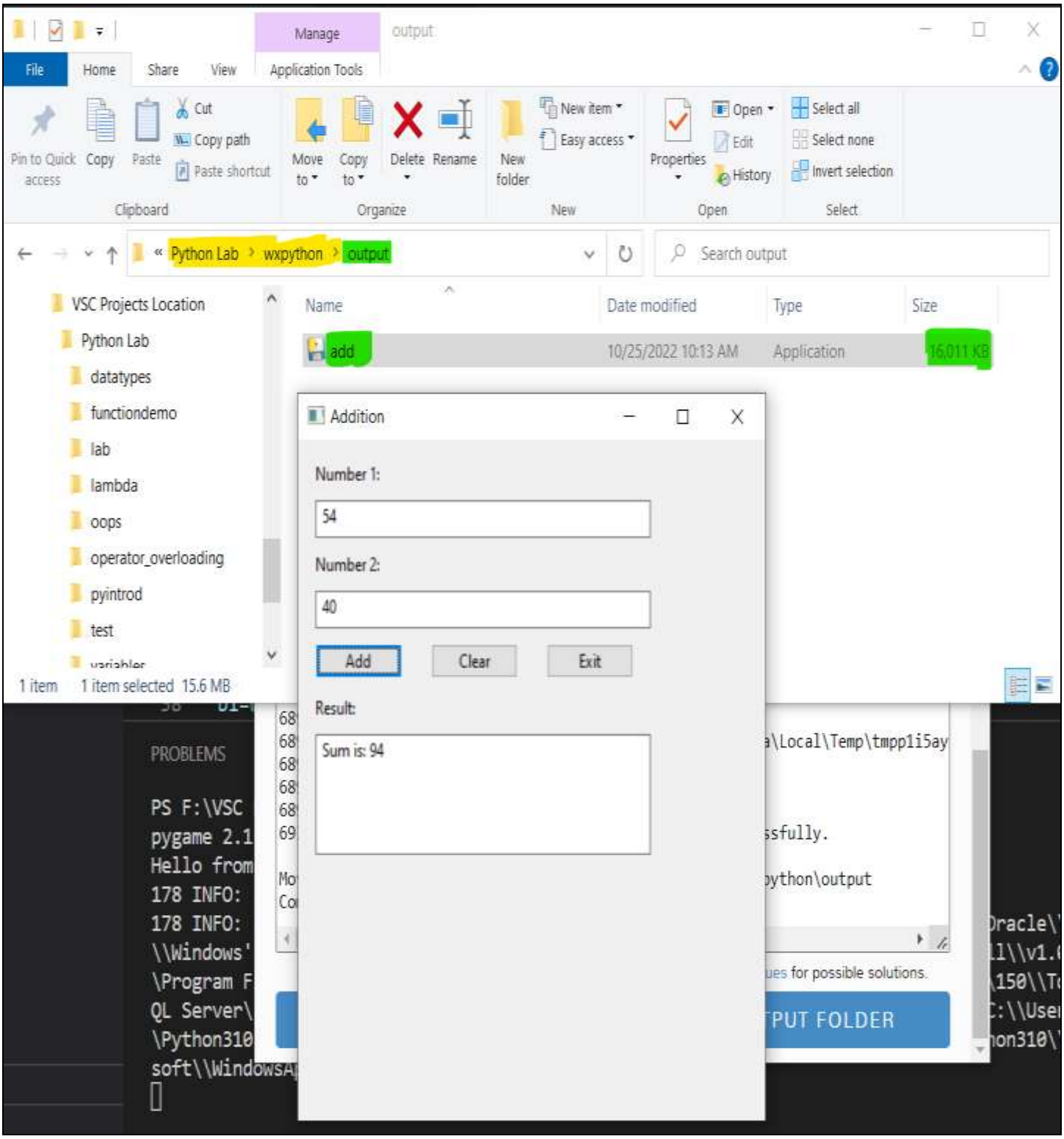
Then press the **CONVERT .PY TO .EXE** button

2.4 PROGRESS AND COMPLETION DETAILS



2.5 VERIFICATION OF EXE FILE

- The target exe file will be stored in the **/output** directory of VSC project location, after the successful execution



2. CONVERT PY TO EXE USING PYINSTALLER

- Pyinstaller is another popular python module which is used to convert the python file (.py) to windows executable file (.exe)
- Here **python program and all of its dependencies are combined into single package** using the pyinstaller module
- Additionally, this module might produce windows, linux and macOS executables

STEPS FOR INSTALLING PYINSTALLER

STEP 1:

- Run the pip command below to install pyinstaller package

```
pip install pyinstaller
```

STEP 2:

- Run the command to convert .py to .exe file

```
pyinstaller –onefile -w <filename.py>
```

Where,

- option **–onefile** will produce **single executable file** (.exe)
- option **-w** means **windows based** which hides the console based (CMD window).

OPTIONS OF PYINSTALLER

S.N	Argument	Value
1.	-F or --onefile	It will create a single executable file (one file bundled executable)
2.	-D or --onedir	• It will create a single folder bundle containing an executable file. • This is the default mode.
3.	--add -data	This option allows you to add data files that need to be bundled with the executable. This option can be applied to multiple times

Syntax

```
pyinstaller --add-data <source; destination>     # windows
pyinstaller --add-data <source: destination>   # linux
```

Example

```
pyinstaller –add-data “info.csv;.” disp.py
```

II. CONVERSION OF PY TO EXE USING PYINSTALLER

Language	:	Python 3
Editor	:	VSC Editor
OS	:	Windows 10
GUI Framework	:	wxPython
Installer	:	**pyinstaller**

1. SOURCE CODE

```
import wx
# create an object for application class
ob=wx.App()
# create a root window
rt=wx.Frame(None, title="Addition", size=(420,490))
# create a panel layout
pl=wx.Panel(rt)
# BUTTON EVENT HANDLER 1
def disp(et):
# get the first input from the user via first text box
    u1=tb1.GetValue()
# convert string formatted number to number
    a=int(u1)
# get the second input from the user via second text box
    u2=tb2.GetValue()
# convert string to integer
    b=int(u2)
# add two numbers
    c=a+b
```

```
# display the result in mult-line text box which accepts only string data
    rs.SetValue("Sum is: "+str(c))
# BUTTON EVENT HANDLER 2
def clearUI(et):
    tb1.SetValue("")
    tb2.SetValue("")
    rs.SetValue("")
# BUTTON EVENT HANDLER 3
def skipApp(et):
    wx.Exit()
# add label 1
l1=wx.StaticText(pl, label="Number 1: ", pos=(15,15))
# add text box 1
tb1=wx.TextCtrl(pl, size=(290,25), pos=(15,40))
# add label 2
l2=wx.StaticText(pl, label="Number 2: ", pos=(15,75))
# add text box 2
tb2=wx.TextCtrl(pl, size=(290,25), pos=(15,100))
# add button 1, 2, 3
b1=wx.Button(pl, label="Add", pos=(15,135))
# add event handler to button 1
b1.Bind(wx.EVT_BUTTON,disp)
b2=wx.Button(pl, label="Clear", pos=(115,135))
# add event handler to button 2
b2.Bind(wx.EVT_BUTTON, clearUI)
# add event handler to button 3
b3=wx.Button(pl, label="Exit", pos=(215,135))
b3.Bind(wx.EVT_BUTTON, skipApp)
```

```
# add label
lb=wx.StaticText(pl, label="Result: ", pos=(15, 170))
# add multi-line text box
rs=wx.TextCtrl(pl, size=(290,80), pos=(15,195))
# activate and display root window
rt.Show()
# show the window in center screen
rt.Centre()
# run the application
ob.MainLoop()
```

2. OUTPUT

2.1 STARTING PYINSTALLER

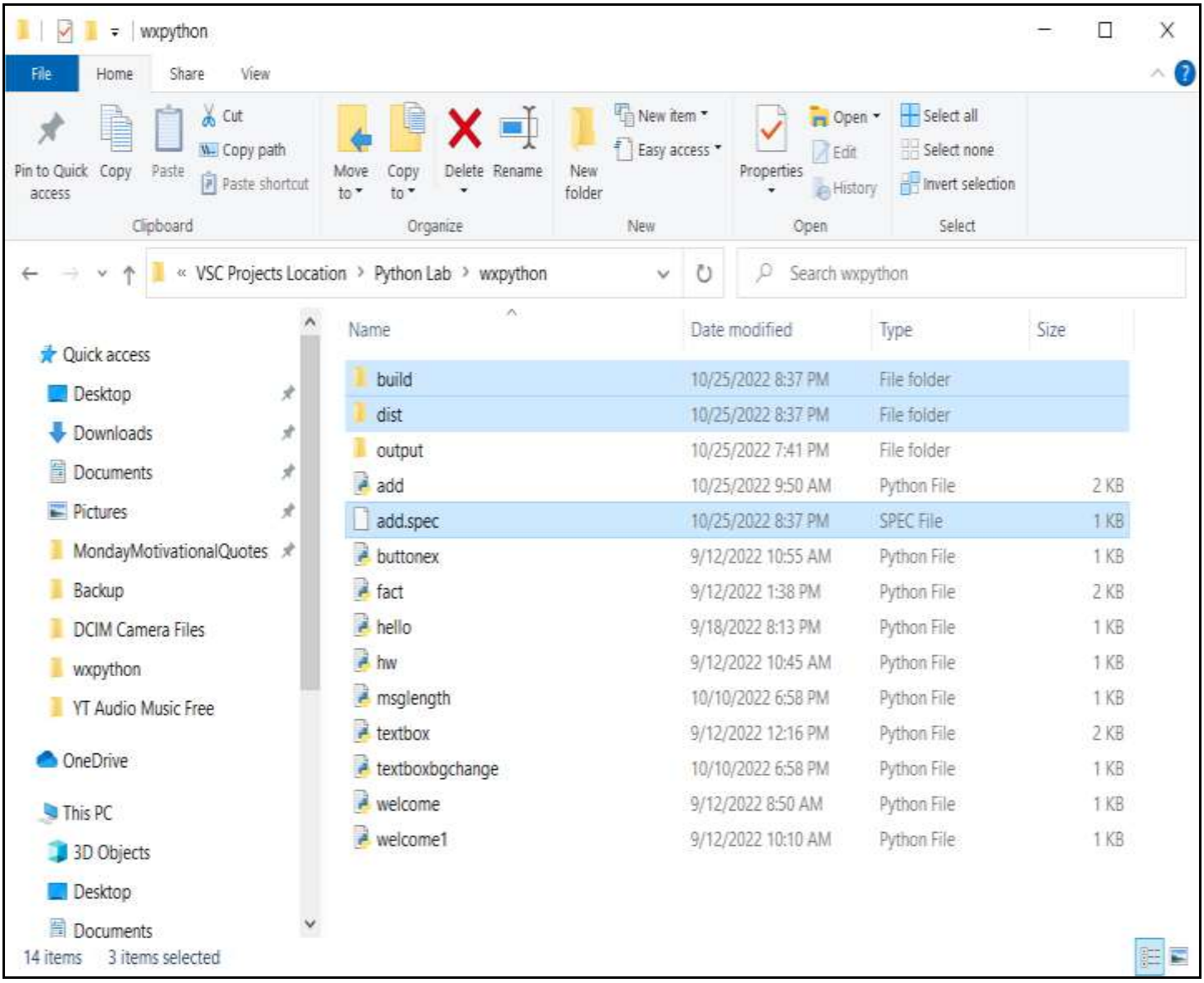
```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    JUPYTER

PS F:\VSC Projects Location\Python Lab\wxpython> pyinstaller --onefile -w add.py
134 INFO: PyInstaller: 5.6
134 INFO: Python: 3.10.5
153 INFO: Platform: Windows-10-10.0.19044-SP0
154 INFO: wrote F:\VSC Projects Location\Python Lab\wxpython\add.spec
157 INFO: UPX is not available.
168 INFO: Extending PYTHONPATH with paths
['F:\\VSC Projects Location\\Python Lab\\wxpython']
pygame 2.1.2 (SDL 2.0.18, Python 3.10.5)
Hello from the pygame community. https://www.pygame.org/contribute.html
1233 INFO: checking Analysis
1234 INFO: Building Analysis because Analysis-00.toc is non existent
1234 INFO: Initializing module dependency graph...
1237 INFO: Caching module graph hooks...
1335 WARNING: Several hooks defined for module 'numpy'. Please take care they do not conflict.
1428 INFO: Analyzing base_library.zip ...
4211 INFO: Loading module hook 'hook-heapq.py' from 'C:\\Users\\Krishna\\AppData\\Local\\Programs\\Py
s\\PyInstaller\\hooks'...
4331 INFO: Loading module hook 'hook-encodings.py' from 'C:\\Users\\Krishna\\AppData\\Local\\Programs
kages\\PyInstaller\\hooks'...
6196 INFO: Loading module hook 'hook-pickle.py' from 'C:\\Users\\Krishna\\AppData\\Local\\Programs\\P
es\\PyInstaller\\hooks'...
8180 INFO: Caching module dependency graph...
8374 INFO: running Analysis Analysis-00.toc
8406 INFO: Adding Microsoft.Windows.Common-Controls to dependent assemblies of final executable
  required by C:\Users\Krishna\AppData\Local\Programs\Python\Python310\python.exe
8501 INFO: Analyzing F:\VSC Projects Location\Python Lab\wxpython\add.py
9334 INFO: Processing module hooks...
9354 INFO: Loading module hook 'hook-_tkinter.py' from 'C:\\Users\\Krishna\\AppData\\Local\\Programs\
```

2.2 CONVERSION RESULT USING PYINSTALLER

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    JUPYTER

19070 INFO: Building EXE from EXE-00.toc
19071 INFO: Copying bootloader EXE to F:\VSC Projects Location\Python Lab\wxpython\dist\add.exe.notanexecuta
19180 INFO: Copying icon to EXE
19180 INFO: Copying icons from ['C:\\Users\\Krishna\\AppData\\Local\\Programs\\Python\\Python310\\lib\\site-
loader\\images\\icon-windowed.ico']
19274 INFO: Writing RT_GROUP_ICON 0 resource with 104 bytes
19275 INFO: Writing RT_ICON 1 resource with 3752 bytes
19276 INFO: Writing RT_ICON 2 resource with 2216 bytes
19277 INFO: Writing RT_ICON 3 resource with 1384 bytes
19277 INFO: Writing RT_ICON 4 resource with 38188 bytes
19278 INFO: Writing RT_ICON 5 resource with 9640 bytes
19278 INFO: Writing RT_ICON 6 resource with 4264 bytes
19279 INFO: Writing RT_ICON 7 resource with 1128 bytes
19282 INFO: Copying 0 resources to EXE
19282 INFO: Embedding manifest in EXE
19284 INFO: Updating manifest in F:\VSC Projects Location\Python Lab\wxpython\dist\add.exe.notanexecutable
19388 INFO: Updating resource type 24 name 1 language 0
19393 INFO: Appending PKG archive to EXE
19418 INFO: Fixing EXE headers
22554 INFO: Building EXE from EXE-00.toc completed successfully.
PS F:\VSC Projects Location\Python Lab\wxpython>
```

2.3 VERIFICATION OF .EXE FILE

- After the successful completion of pyinstaller, two directories like **build**, **dist** and one file **<file>.spec** will be created in the target path.
- The executable file will be stored in the **/dist** folder



1. build

- This folder contains associated log files, working files and other additional files needed by pyinstaller

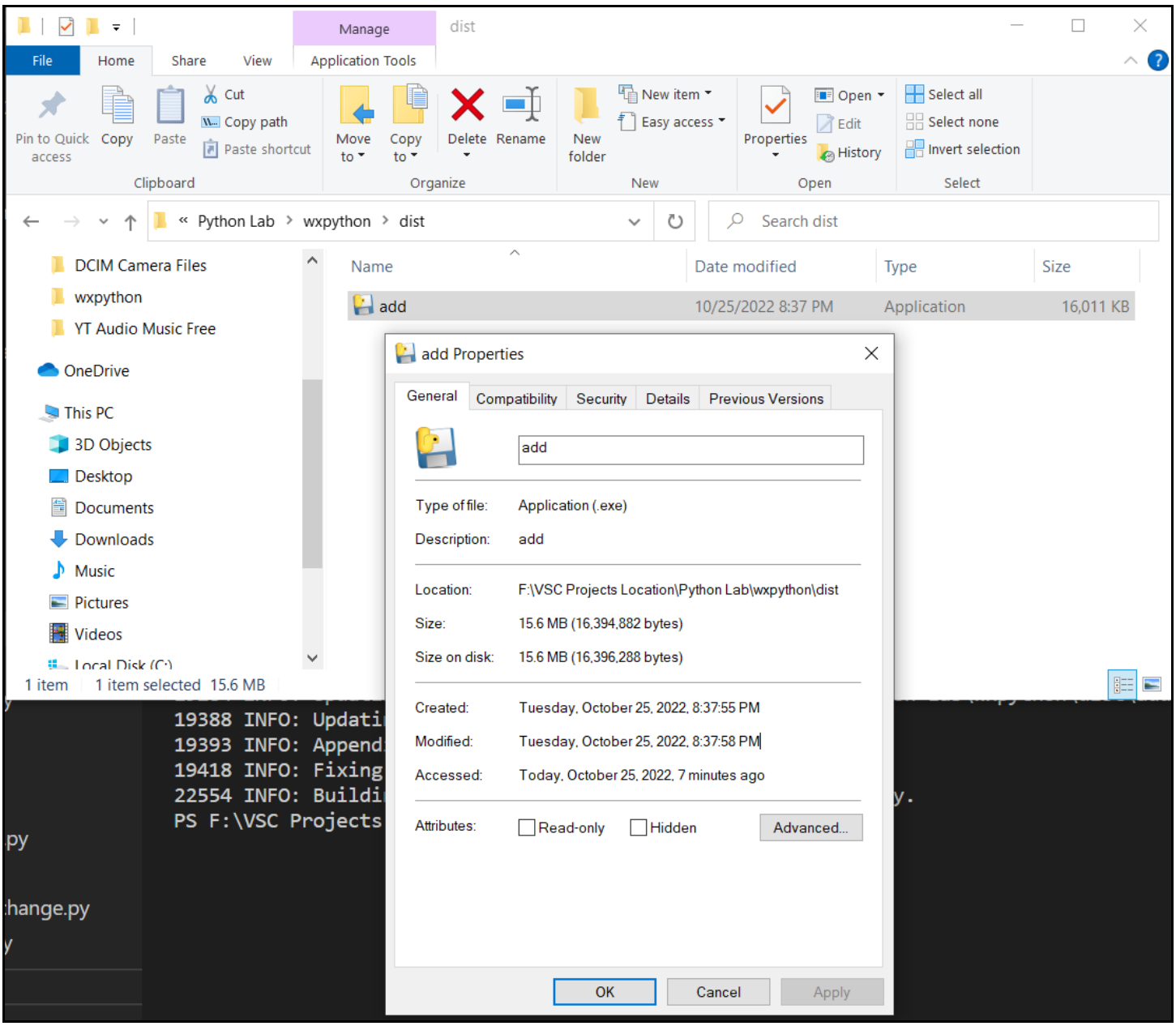
2. dist

- This folder **contains the distributable version of the python script**
- This is the folder contains executable file that we can share to others

3. <file>.spec

- It has the same name as the python file
- This is the specification file which contains the configuration information like script name, dependent libraries, data files, etc,..

2.3 VERIFICATION OF .EXE FILE - CONTINUE



2.4 RUNNING GUI EXECUTABLE