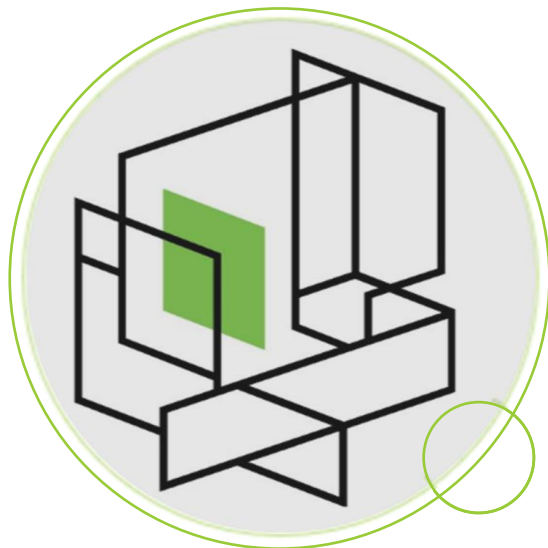


## TABLE OF CONTENTS

Launching the Application from Command Line .....	3
SFTPPipeline Dashboard .....	4
Active Upload Widget .....	5
Admin Login Widget.....	6
Object Management:.....	7
QtDesigner: Installation & Main Window .....	10
Adding UI Files to the Python Application .....	12
Packaging Into An Executable .....	13
Database Schema.....	15



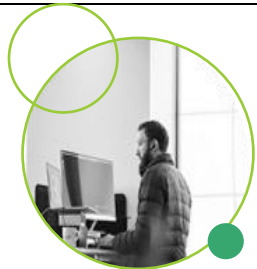
# SFTP Pipeline User Guide



Urban Science

## EXTRA FUNCTIONALITY

### QtDesigner | Pyuic5



QtDesigner .ui can be loaded directly by the constructor or converted to .py files using the following command, as discussed in the **pyuic5** section:  
*pyuic5 -x example.ui -o example.py*

### PyQt5-tools



Contains QtDesigner. Install with:  
*pip install pyqt5-tools*  
Located in your python folder:  
*[python folder] / Lib / site-packages / qt5\_applications / Qt / bin/ designer.exe*

### PyInstaller

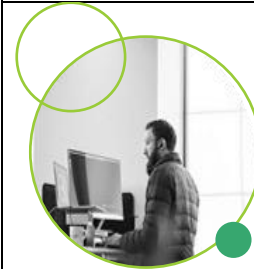


Allows packaging the SFTP tool into a single executable. Install with:  
*pip install pyinstaller*

5

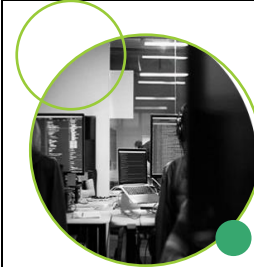
## MODULE DEPENDENCIES

### PyQt5



GUI framework in which the User Interface is developed. Can be used to implement new UI elements in future development.

### MySQL Connector



Self-contained python driver for communicating with MySQL servers.

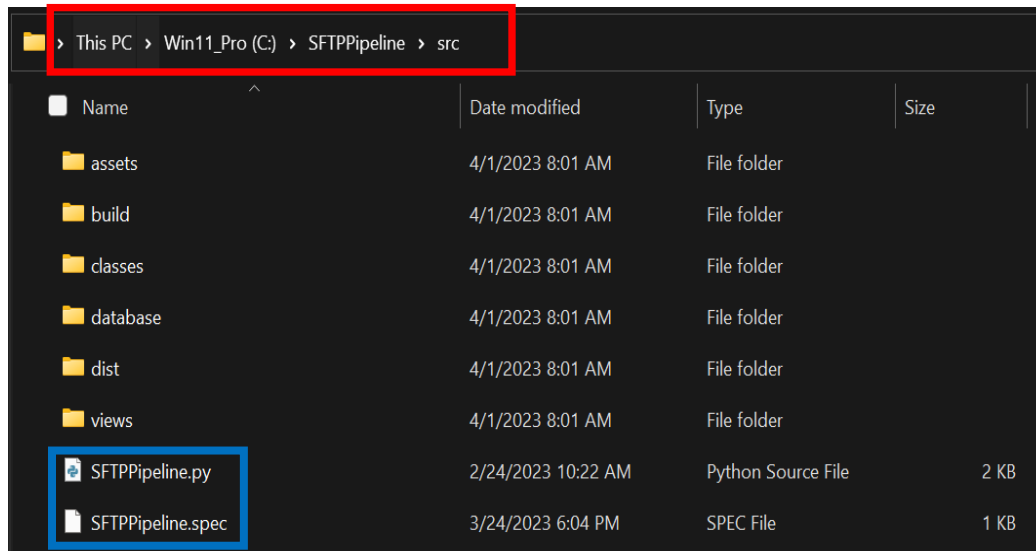
### Hash Lib



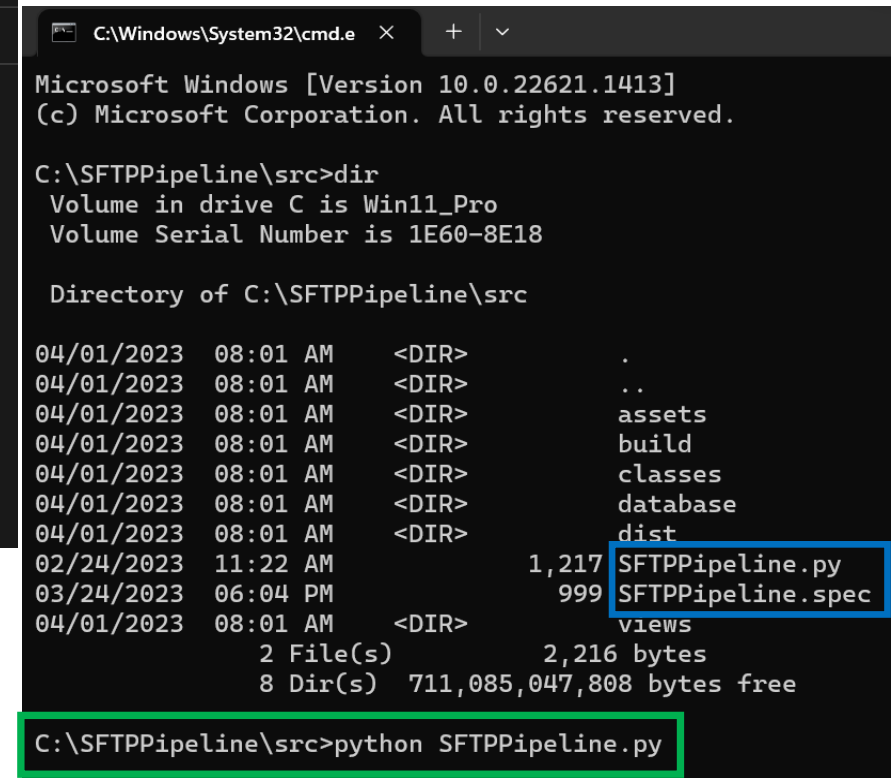
Contains cryptographic and hashing functions that are used in the application.

## LAUNCHING THE APPLICATION FROM COMMANDLINE

Open your File Explorer, and navigate to the 'application/src' directory...



Inside the '/src' directory, you see the SFTPPipeline.py file.  
Open a Terminal window in this location, and then execute the following command: `'python SFTPPipeline.py'`



The application window should appear.

Note: Errors may occur in the absence of modules the application depends upon, and may be reconciled with the execution of the following command:

`'pip install [API/Module Name]'`

## SFTPPipeline Dashboard

The SFTPPipeline Dashboard initially consists of 5 main widgets:

- A.** Displays user-created **Jobs** consisting of one or more **Presets**
- B.** Displays user-created **Presets** consisting of one or more **Files**
- C.** Displays connections to any number of SFTP servers
- D.** Allows authorized users to navigate to the **Administrative Login** portal.
- E.** **Active Uploads:** Each **Job/Presets** will display upload progress with respect to individual files.

The screenshot shows the SFTPPipeline Dashboard interface. At the top, there is a header bar with the Urban Science logo and the text "URBAN SCIENCE®". To the right of the logo is a link labeled "Admin" with a red circle "D" next to it. Below the header, there are three main widgets: "Jobs" (labeled with a red circle "A"), "Presets" (labeled with a red circle "B"), and "SFTP Connections" (labeled with a red circle "C"). The "Jobs" widget is currently empty. The "Presets" widget displays a list of three test presets: "Test Preset 1", "Test Preset 2", and "Test Preset 3". The "SFTP Connections" widget displays a table with two columns: "Server" and "Remote Directory". The table contains one entry: "1 sftp.alexpro.net" and "sftp". Below these widgets is a section labeled "Active Uploads" (labeled with a red circle "E"). This section has a table with five columns: "File", "SFTP Server", "Remote Directory", "Preset", and "Progress". The table is currently empty. To the right of the table is a "Clear" button. At the bottom of the dashboard, there is a green bar with an "UPLOAD" button.

Server	Remote Directory
1 sftp.alexpro.net	sftp

File	SFTP Server	Remote Directory	Preset	Progress
------	-------------	------------------	--------	----------

UPLOAD

## Active Uploads Widget

Consider the **Active Uploads** widget below.

	File	SFTP Server	Remote Directory	Preset	Progress
1	C:/Users/Aquarii/Desktop...	sftp.alexpro.net	sftp	Test Preset 1	0%
2	C:/Users/Aquarii/Desktop...	sftp.alexpro.net	sftp	Test Preset 1	0%
3	C:/Users/Aquarii/Desktop...	sftp.alexpro.net	sftp	Test Preset 1	0%
4	C:/Users/Aquarii/Desktop...	sftp.alexpro.net	sftp	Test Preset 3	0%
5	C:/Users/Aquarii/Desktop...	sftp.alexpro.net	sftp	Test Preset 3	0%

Deliverables can be selected in the form of **Jobs** and **Presets**.

Multiple **Jobs/Presets** can be selected simultaneously, and their individual files will be added to the **Active Uploads** pane.

Upon clicking **Upload**, their progress is displayed incrementally, with respect to individual files.

**Important Note:** New **Users**, **Presets**, **Jobs**, and **SFTP Connections** can only be created by authorized users who have been authenticated through the **Admin Login** widget, as depicted in the following section.

## Admin Login Widget

URBAN SCIENCE®

**Admin**

Presets

Test Preset 1
Test Preset 2
Test Preset 3

SFTP Connections

Server	Remote Directory
1 sftp.alexpro.net	sftp

Clear

Authorized users can click the **Admin** button in the top right corner of their dashboard.

The environment depicted in this documentation uses example credentials, in which the passwords have been hashed and salted.

URBAN SCIENCE® Home

Please Log In

Username

Steven

Password

\*\*\*\*\*

Log In

The window will change to display an administrative login form as depicted above.

Users can return to the previous window by clicking the **Home** button in the upper right region.

URBAN SCIENCE® Home

Users Presets Jobs Connections

SFTP Connections

Server	Remote Directory
1 sftp.alexpro.net	sftp

Delete Selected Connection(s)

New Connection

Server

Username

Password

Remote Directory

Add Connection

The following window possesses multiple Tabs, allowing **Admin** users greater control.

Each of these tabs allow users to create, manage, and delete **Users**, **Presets**, **Jobs**, and **Connection** objects from the online database; affecting what other users see and have access to.

## Object Management: Users List

Users

Presets

Jobs

Connections

Users List

Alex

Jordan

Gabriel

Yoshi

Blaine

Steven

Delete Selected User(s)

New User

Username

Password

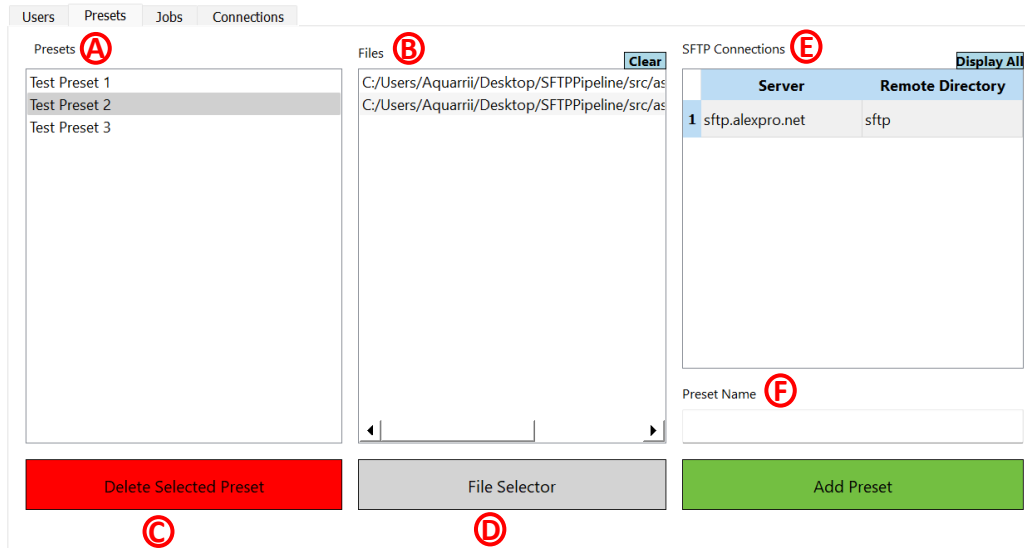
Add User

The **Users List** allows Admins to add and delete Users from the database.

In this example, this affects Users' ability to use the Administrative widgets

## Object Management: Presets List

The Presets List has a considerably more intricate interface.



Ⓒ **Presets** can be effortlessly deleted by selecting one or more **Presets** in the **Presets** pane Ⓐ followed by clicking the button **Delete Selected Preset**. Ⓒ

Ⓐ Existing **Presets** can be selected in this window, causing the associating files to populate the **Files List** Ⓑ.

Ⓓ New **Presets** can be created by clicking the **File Selector** window opens a new window. users can then select multiple files to add to the **Files List**.

Ⓔ After doing so, one or more **Servers** Ⓔ can be selected, designating them to the **Preset**. Click a server a 2<sup>nd</sup> time if you wish to deselect it and disassociate it from the **Preset** you are creating.

Ⓕ Lastly, a **Preset Name** must be provided in the insertion field, and the **Add Preset** button clicked in order for the new **Preset** to be created.



## Object Management: Connections List

Users

Presets

Jobs

Connections

SFTP Connections

	Server	Remote Directory
1	sftp.alexpro.net	sftp

Delete Selected Connection(s)

New Connection

Server

Username

Password

Remote Directory

Add Connection

Users can create new **Connections** from this window, by providing the following, and then clicking **Add Connection**

- A **Server Name**
- A **Username**
- A **Password**
- A valid URL for the **Remote Directory**.

**Connections** can be removed by selecting them in the list and then clicking **Delete Selected Connection(s)**.

## QtDesigner: Installation

The first step is installing QtDesigner. Open a Command Prompt and execute the command

```
pip install pyqt5-tools
```

You can find your python folder by executing the command:

```
python -c "import sys; print(sys.path)"
```

Then navigating to the folder:

```
... / [Pythonxxx] / Lib / site-packages /  
qt5_applications / Qt / bin/ designer.exe
```

Right-click your executable and **Pin To Start**, **Pin to Taskbar**, or **Create [a] Shortcut** to save yourself time in the future.

Launch the application.

```
C:\Windows\System32>pip install pyqt5-tools
Collecting pyqt5-tools
  Downloading pyqt5_tools-5.15.9.3.3-py3-none-any.whl (29 kB)
Requirement already satisfied: pyqt5==5.15.9 in c:\users\steve\appdata\local\programs\python\python310\lib\site-packages (from pyqt5-tools) (5.15.9)
Collecting python-dotenv
  Downloading python_dotenv-1.0.0-py3-none-any.whl (19 kB)
Collecting click
  Downloading click-8.1.3-py3-none-any.whl (96 kB)
-----
96.6/96.6 kB 262.5 kB/s eta 0:00:00
Collecting pyqt5-plugins<5.15.9.3,>=5.15.9.2.2
  Downloading pyqt5_plugins-5.15.9.2.3-cp310-cp310-win_amd64.whl (66 kB)
-----
66.8/66.8 kB 725.2 kB/s eta 0:00:00
Requirement already satisfied: PyQt5-Qt5>=5.15.2 in c:\users\steve\appdata\local\programs\python\python310\lib\site-packages (from pyqt5==5.15.9->pyqt5-tools) (5.15.2)
Requirement already satisfied: PyQt5-sip<13,>=12.11 in c:\users\steve\appdata\local\programs\python\python310\lib\site-packages (from pyqt5==5.15.9->pyqt5-tools) (12.11.1)
Collecting qt5-tools<5.15.2.2,>=5.15.2.1.2
  Downloading qt5_tools-5.15.2.1.3-py3-none-any.whl (13 kB)
Collecting colorama
  Downloading colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Collecting qt5-applications<5.15.2.3,>=5.15.2.2.2
  Downloading qt5_applications-5.15.2.2.3-py3-none-win_amd64.whl (64.5 MB)
-----
64.5/64.5 MB 20.4 MB/s eta 0:00:00
Installing collected packages: qt5-applications, python-dotenv, colorama, click, qt5-tools, pyqt5-plugins, pyqt5-tools
Successfully installed click-8.1.3 colorama-0.4.6 pyqt5-plugins-5.15.9.2.3 pyqt5-tools-5.15.9.3.3 python-dotenv-1.0.0 qt5-applications-5.15.2.2.3 qt5-tools-5.15.2.1.3
```

steve > AppData > Local > Programs > Python > Python310 > Lib > site-packages > qt5\_applications > Qt > bin

Name	Date modified	Type	Size
assistant.exe	4/5/2023 6:42 PM	Application	870 KB
canbusutil.exe	4/5/2023 6:42 PM	Application	48 KB
concr140.dll	4/5/2023 6:42 PM	Application extension	320 KB
designer.exe	4/5/2023 6:42 PM	Application	550 KB
dumpcpp.exe	4/5/2023 6:42 PM	Application	206 KB

[ This does not require a visual demonstration. ]

## QtDesigner: Main Window

The main window should open with a smaller, **New Form** window displayed on top of it. The primary two elements you will likely be concerned with are of course **Main Window** and **Widget**. You will first want to create a **Main Window**.

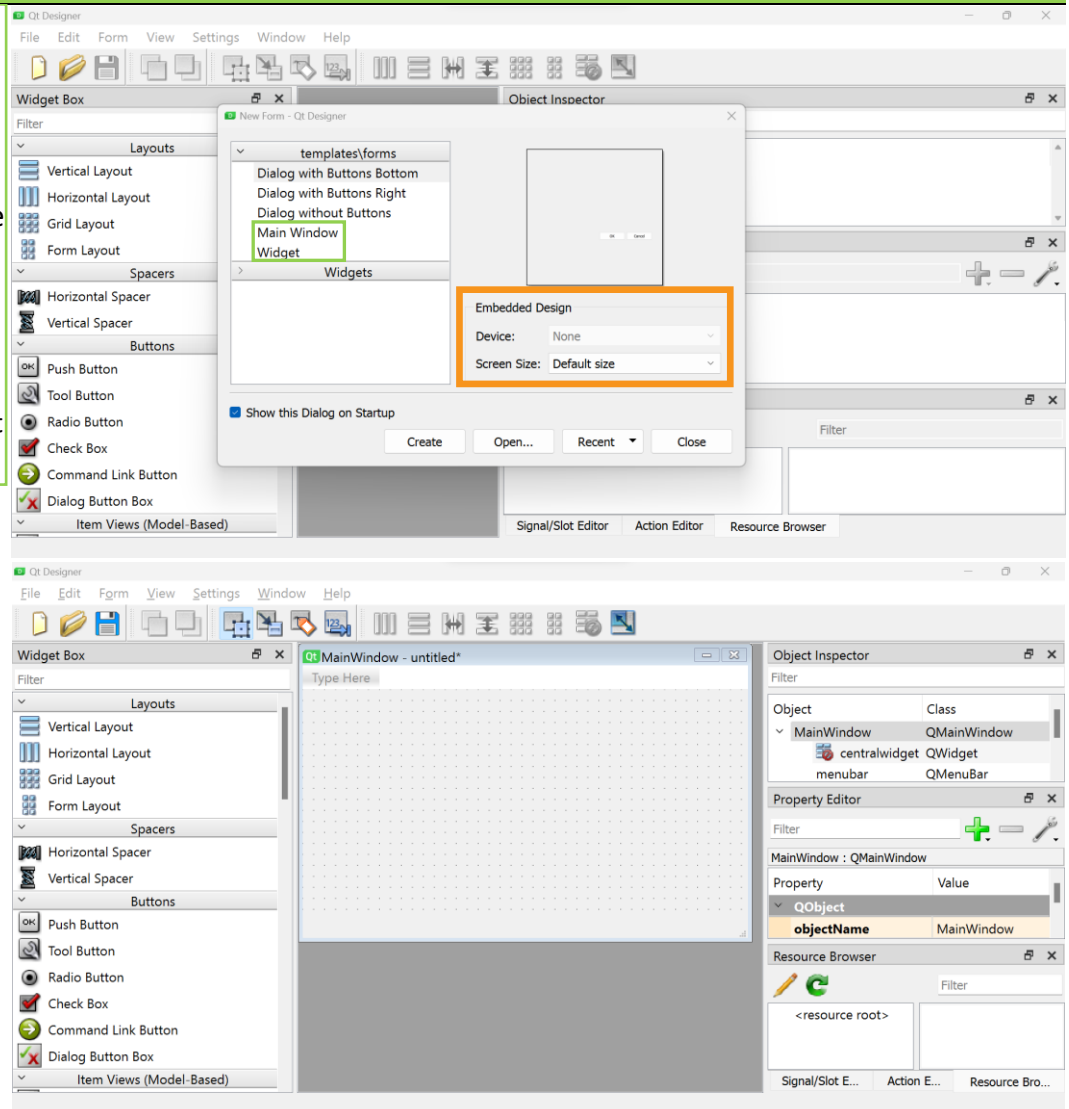
Note: Keep the **Device** and **Screen Size** properties in mind when designing UI windows and widgets, as they may impact performance on a diverse variety of machines.

QtDesigner's visibility may depend on your device's resolution. The new **MainWindow** will appear in the middle pane.

From here, you can insert **Buttons**, **Labels**, **Widgets**, and a myriad of other objects to your Main Window from your **Widget Box** (left). Any object created can be selected via the **Object Inspector** (top right), and its properties viewed and edited in the **Property Editor** further down.

For a more in-depth tutorial on QtDesigner creating complex graphical user interfaces, consider watching the PyQt5 QtDesigner tutorial linked below:

<https://www.youtube.com/playlist?list=PLzMcbGfZo4-IB8MZfHPLTEHO9zJDDLpYj>



## QtDesigner: PYUIC5

~/SFTPPipeline/src/views/MainWindow.py

### OPTIONAL FEATURES

```
24 class MainWindow(QtWidgets.QMainWindow):
25     # Function: Constructor
26     # Parameters: None
27     # Return: MainWindow UI object
28     # Loads the MainWindow.UI file and initializes all of the widget logic
29     def __init__(self):
30         # Call QMainWindow Constructor
31         super(MainWindow, self).__init__()
32         # Load MainWindow.UI
33         uic.loadUi(r"./views/MainWindow.ui", self)
34         # Initialize User
35         self.User = User()
36         # Initialize Security Object
37         self.Security = Security()
38         # Load Widgets
39         self.Admin_Button = self.findChild(QtWidgets.QLabel, "Admin_Button")
40         self.Stacked_Widget = self.findChild(QtWidgets.QStackedWidget, "stackedWidget")
```

```
C:\SFTPPipeline\src\practice_example\views>dir
```

```
Volume in drive C is Win11_Pro
```

```
Volume Serial Number is 1E60-8E18
```

```
Directory of C:\SFTPPipeline\src\practice_example\views
```

```
04/05/2023  08:34 PM    <DIR>          .
04/05/2023  08:34 PM    <DIR>          ..
04/05/2023  08:36 PM               36,273 MainWindow.py
03/27/2023  05:18 PM               40,108 MainWindow.ui
02/13/2023  04:22 PM               11,330 MainWindow_OLD.py
04/05/2023  08:34 PM    <DIR>          __pycache__
                3 File(s)          87,711 bytes
                3 Dir(s)  701,989,183,488 bytes free
```

```
C:\SFTPPipeline\src\practice_example\views>pyuic5 -x MainWindow.ui -o MainWindow.py
```

As mentioned on pg. 2, line 33 of our code demonstrates the python constructor of our code incorporates .ui files into the application directly.

However, the API **pyuic5** can optionally convert .ui files to .py files which can then be used as **Views** in your python application.

## Packaging Into An Executable

You can install PyInstaller with:  
***pip install pyinstaller***

```
C:\SFTPPipeline\src>pip install pyinstaller
Collecting pyinstaller
  Downloading pyinstaller-5.10.1-py3-none-win_amd64.whl (1.3 MB)
    1.3/1.3 MB 8.0 MB/s eta 0:00:00
Requirement already satisfied: setuptools>=42.0.0 in c:\users\steve\appdata\local\programs\python\python310\lib\site-packages (from pyinstaller) (65.5.0)
Collecting altgraph
  Downloading altgraph-0.17.3-py2.py3-none-any.whl (21 kB)
Collecting pywin32-ctypes>=0.2.0
  Downloading pywin32-ctypes-0.2.0-py2.py3-none-any.whl (28 kB)
Collecting pefile>=2022.5.30
  Downloading pefile-2023.2.7-py3-none-any.whl (71 kB)
    71.8/71.8 kB ? eta 0:00:00
Collecting pyinstaller-hooks-contrib>=2021.4
  Downloading pyinstaller_hooks_contrib-2023.2-py2.py3-none-any.whl (261 kB)
    262.0/262.0 kB 16.8 MB/s eta 0:00:00
Installing collected packages: pywin32-ctypes, altgraph, pyinstaller-hooks-contrib, pefile, pyinstaller
Successfully installed altgraph-0.17.3 pefile-2023.2.7 pyinstaller-5.10.1 pyinstaller-hooks-contrib-2023.2 pywin32-ctypes-0.2.0
```

Then, navigate to your ‘~/SFTPPipeline/src ‘ folder:

This PC > Win11_Pro (C:) > SFTPPipeline > src		
Name	Date modified	Type
assets	4/1/2023 8:01 AM	File folder
build	4/1/2023 8:01 AM	File folder
classes	4/5/2023 9:36 PM	File folder
database	4/5/2023 9:36 PM	File folder
dist	4/17/2023 11:49 AM	File folder
views	4/5/2023 9:36 PM	File folder
SFTPPipeline.py	2/24/2023 10:22 AM	Python Source File
SFTPPipeline.spec	4/17/2023 11:48 AM	SPEC File

## Execute the following command:

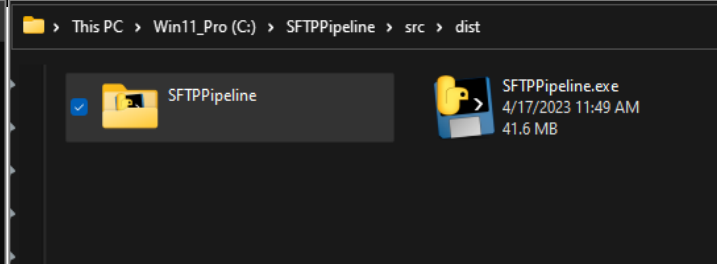
```
pyinstaller --add-data "views;views" --add-data "assets;assets" --onefile SFTPPipeline.py
```

A very intimidating wall of output should flood your command line.

## Note:

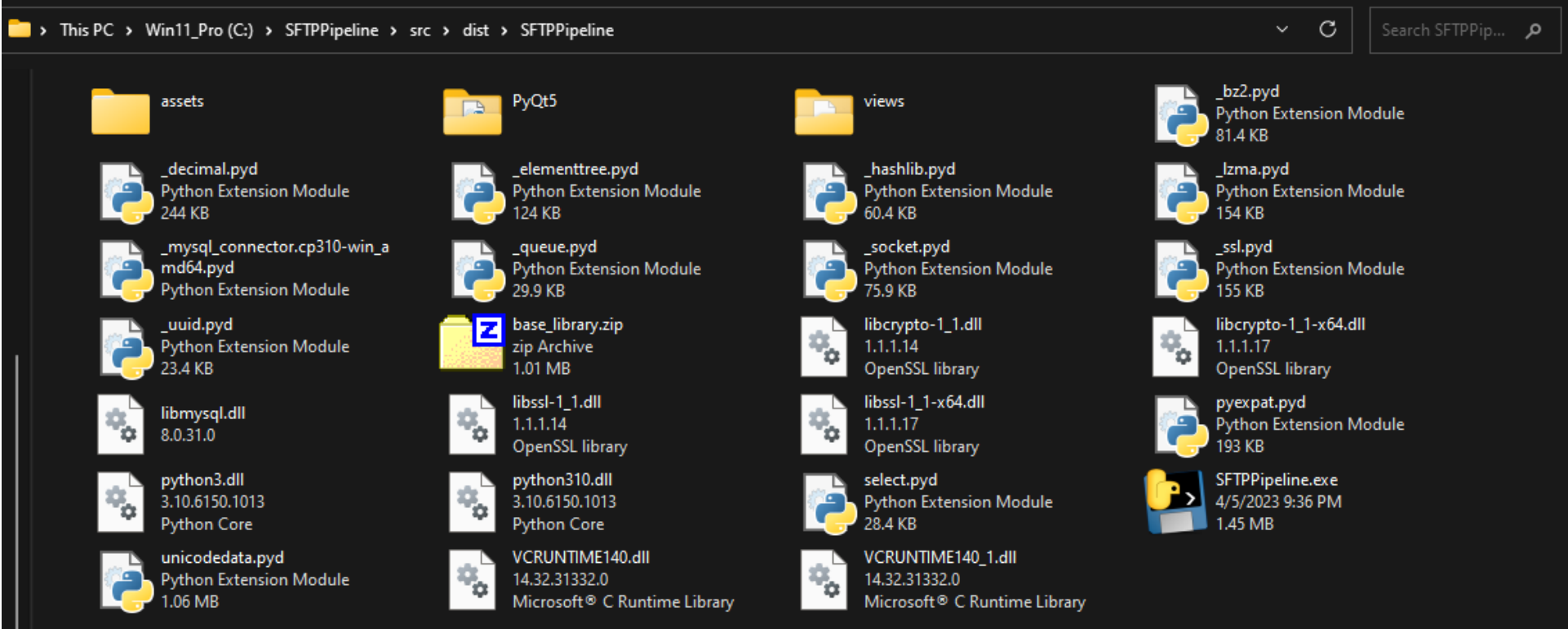
*Pyinstaller will produce the folder and executable depicted below. Ignore the external executable. The SFTPPipeline folder contains all the assets, as well as the EXE it needs.*

```
C:\Windows\System32\cmd.e  x  +  v
C:\SFTPPipeline\src>pyinstaller --add-data "views;views" --add-data "assets;assets" --onefile SFTPPipeline.py
693 INFO: PyInstaller: 5.10.1
693 INFO: Python: 3.10.10
705 INFO: Platform: Windows-10-10.0.22621-SP0
707 INFO: wrote C:\SFTPPipeline\src\SFTPPipeline.spec
711 INFO: UPX is not available.
718 INFO: Extending PYTHONPATH with paths
['C:\\SFTPPipeline\\src']
1167 INFO: Appending 'datas' from .spec
1171 INFO: checking Analysis
1184 INFO: Building because inputs changed
1184 INFO: Initializing module dependency graph...
1187 INFO: Caching module graph hooks...
1220 INFO: Analyzing base_library.zip ...
3516 INFO: Loading module hook 'hook-encodings.py' from 'C:\\Users\\steve\\AppData\\Local\\Programs\\Python\\Python310\\
kages\\PyInstaller\\hooks'...
5664 INFO: Loading module hook 'hook-pickle.py' from 'C:\\Users\\steve\\AppData\\Local\\Programs\\Python\\Python310\\
es\\PyInstaller\\hooks'...
6828 INFO: Loading module hook 'hook-heapq.py' from 'C:\\Users\\steve\\AppData\\Local\\Programs\\Python\\Python310\\
s\\PyInstaller\\hooks'...
8546 INFO: Caching module dependency graph...
8689 INFO: running Analysis Analysis-00.toc
8730 INFO: Adding Microsoft.Windows.Common-Controls to dependent assemblies of final executable
required by C:\\Users\\steve\\AppData\\Local\\Programs\\Python\\Python310\\python.exe
9391 INFO: Analyzing C:\\SFTPPipeline\\src\\SFTPPipeline.py
9418 INFO: Loading module hook 'hook-PyQt5.py' from 'C:\\Users\\steve\\AppData\\Local\\Programs\\Python\\Python310\\
s\\PyInstaller\\hooks'...
9691 INFO: Loading module hook 'hook-PyQt5.uic.py' from 'C:\\Users\\steve\\AppData\\Local\\Programs\\Python\\Python310\\
kages\\PyInstaller\\hooks'...
9949 INFO: Processing pre-find module path hook PyQt5.uic.port_v2 from 'C:\\Users\\steve\\AppData\\Local\\Programs\\Python\\Python310\\
\\lib\\site-packages\\PyInstaller\\hooks\\pre_find_module_path\\hook-PyQt5.uic.port_v2.py'.
```



## Database Schema

You can see several folders and files in this directory, including the SFTPPipeline.exe file that will launch this program. This 'SFTPPipeline' folder is self-containing. It can be moved out of the '/src/dist' folder, and into another location, such as moving it to the C drive, such that its location would become 'C:\Program Files\SFTPPipeline'.



Note that this executable successfully launches the application's main page so long as the user has a working connection to their company's database. For the sake of this example, the tool is connecting to a MySQL database for testing purposes.

## Database Schema

