How to Create an Executable for a Python + Qt Project in Qt Creator

1. Install PyInstaller in your Python Environment

Ensure that PyInstaller is installed in your Python virtual environment. You can install it using pip:

1. Open your terminal or command prompt.

2. Activate your virtual environment:

C:\Users\khana\OneDrive\Desktop\3K04\QT_testing\test_windowUI\.qtcreator\Python_3_12_6venv\Scripts\activate

3. Install PyInstaller:

pip install pyinstaller

2. Generate the Executable

After installing PyInstaller, use the following steps to create an executable from your Python script:

1. Navigate to your project directory where the main Python script (widget.py) is located:

cd C:\Users\khana\OneDrive\Desktop\3K04\QT_testing\test_windowUI

2. Run PyInstaller with the following command:

pyinstaller --onefile widget.py

This command will:

- Generate a single executable (--onefile) without dependencies as separate files.

- Create a dist folder containing the widget.exe file.

3. Handling Dependencies (UI, Credentials, etc.)

Since you are using Qt Creator and external resources (like .ui files and credentials.txt), you need to include them in the package. You can specify additional files or directories with the --add-data

option.

Example command:

pyinstaller --onefile widget.py --add-data "ui_form.py;." --add-data "ui_home.py;." --add-data "credentials.txt;."

- --add-data "file_path;destination": This copies external files to the correct location. Here, the credentials.txt file and the UI files need to be included in the same folder as the executable.
- The ;. indicates that the files should be copied to the root folder where the executable resides.

4. Testing the Executable

After PyInstaller finishes, you'll find the executable in the dist folder:

C:\Users\khana\OneDrive\Desktop\3K04\QT_testing\test_windowUI\dist\widget.exe

Try running the executable by double-clicking it or launching it from the command line:

dist\widget.exe

5. Adjust PyInstaller Configuration (if needed)

- If you encounter missing modules or errors related to hidden imports, you can modify the PyInstaller .spec file.
- Run PyInstaller once to generate the .spec file:

pyinstaller widget.py

- Modify the .spec file to include any hidden imports or additional files you need.

6. Distribute the Executable

After testing, you can distribute the widget.exe file to others. The resulting executable will contain all dependencies, so the end user won't need to install Python, PySide6, or any other library.

Additional Tips

- Icon: To add a custom icon to your executable, use the --icon flag:

  pyinstaller --onefile --icon=your_icon.ico widget.py

- Console/No Console: If you don't want a console window to appear when running the executable, add the --noconsole option:

  pyinstaller --onefile --noconsole widget.py