

Developing Cross Platform Application using PyQt5

XIAOJUN PAN

2017/12/21

Python Application

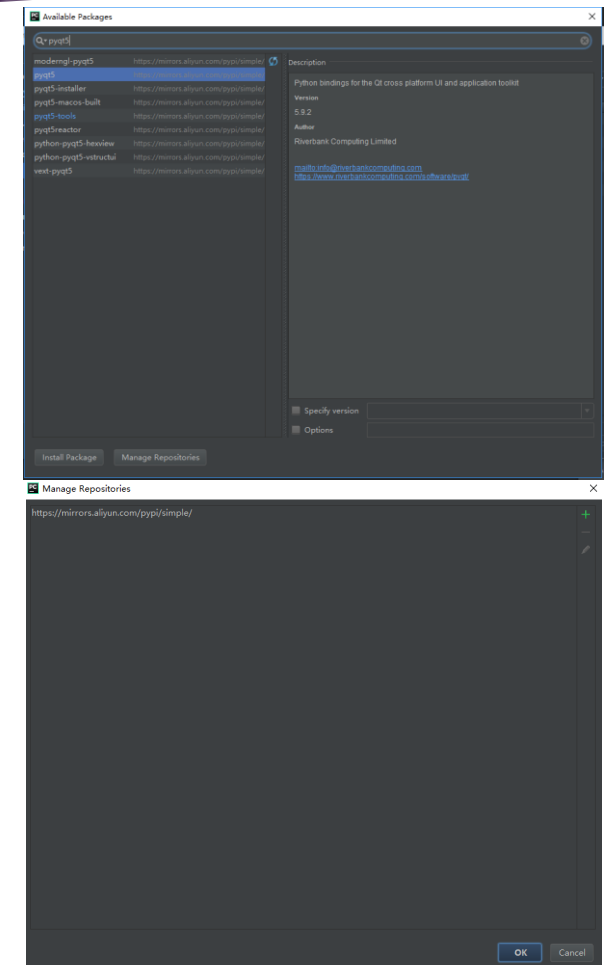
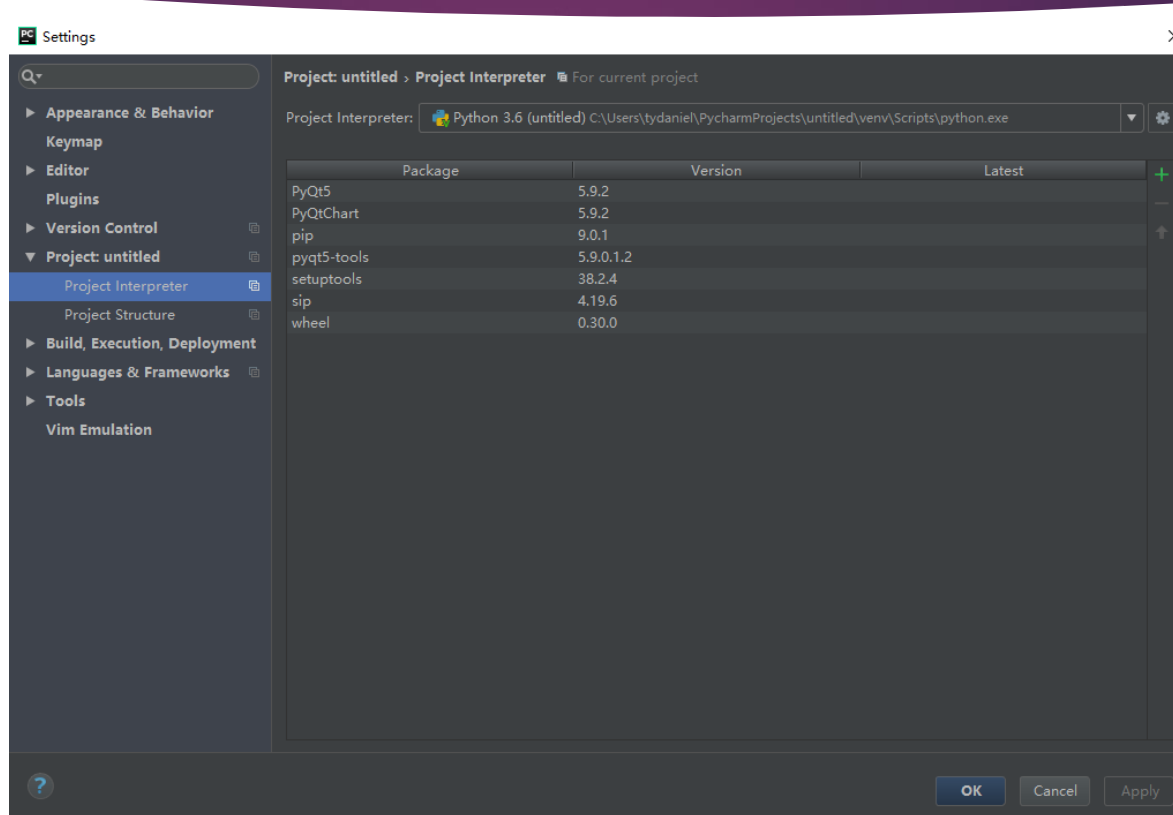
PREPARATION

Preparation

Install Project Interpreter

Interpreters:

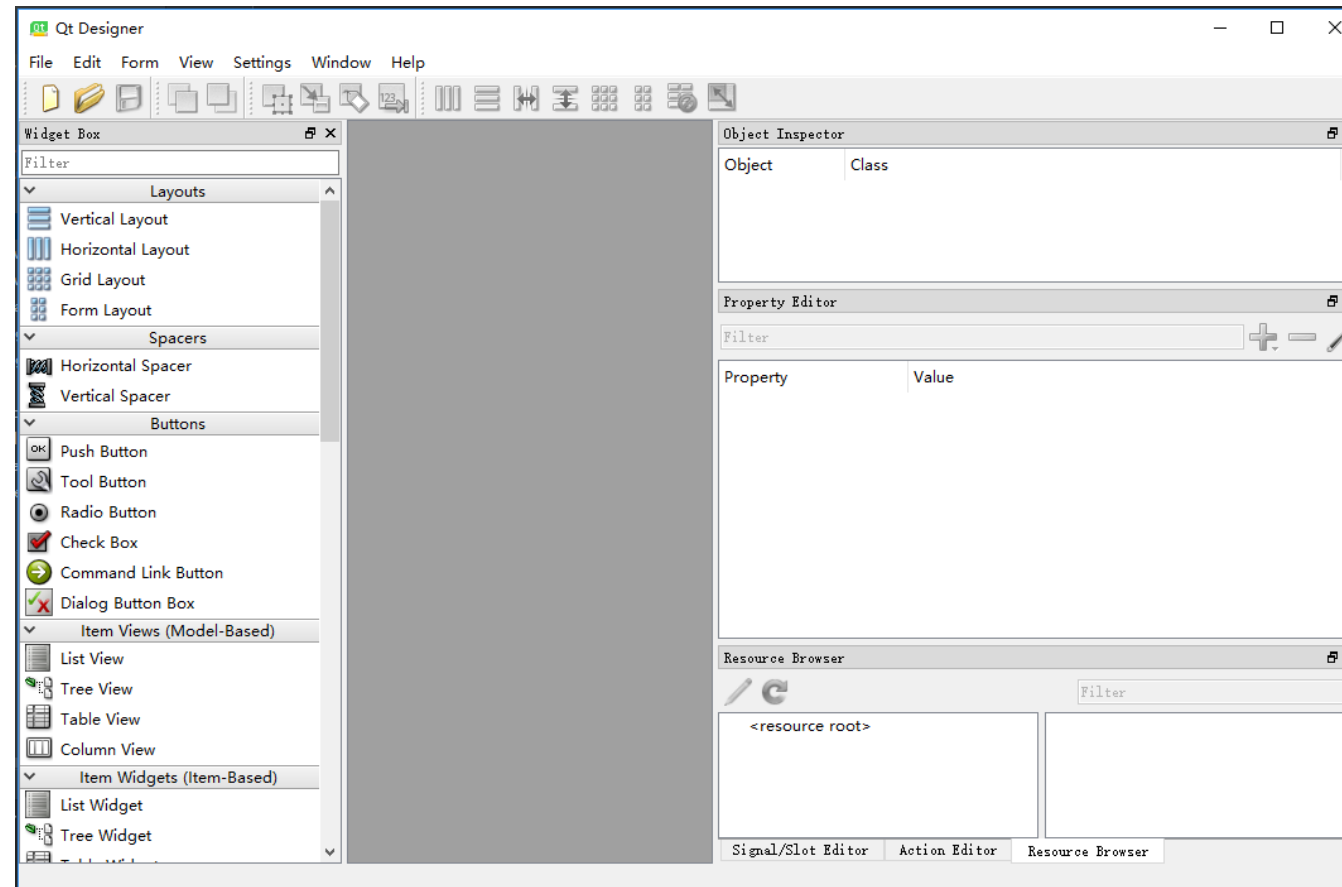
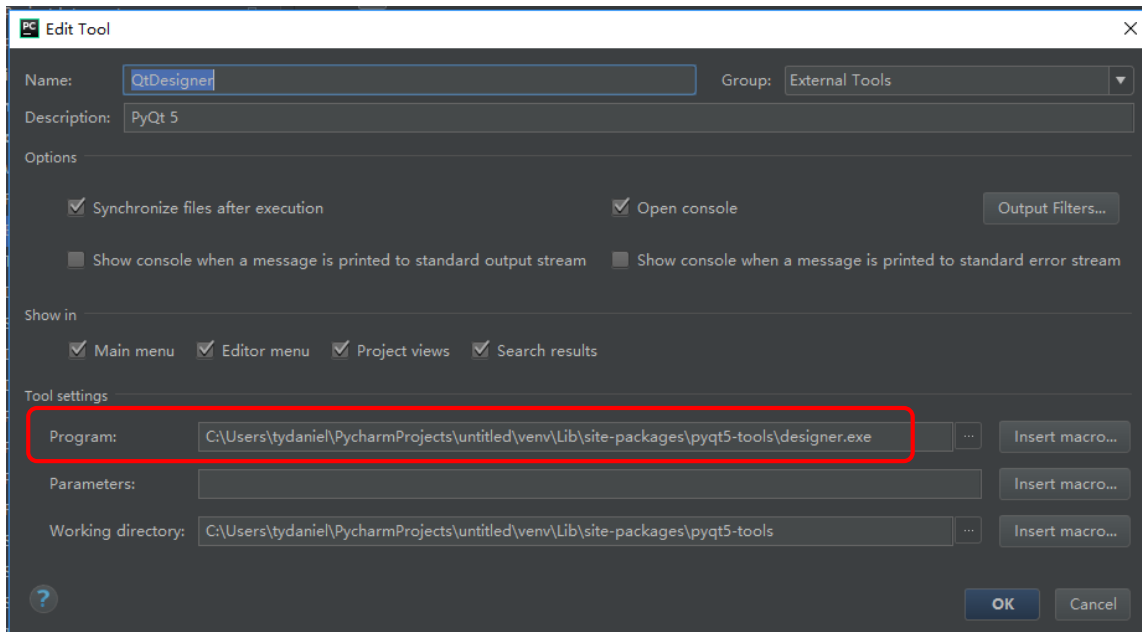
- PyQt5
- Pyqt5-tools
(Qt Designer)
- PyInstaller



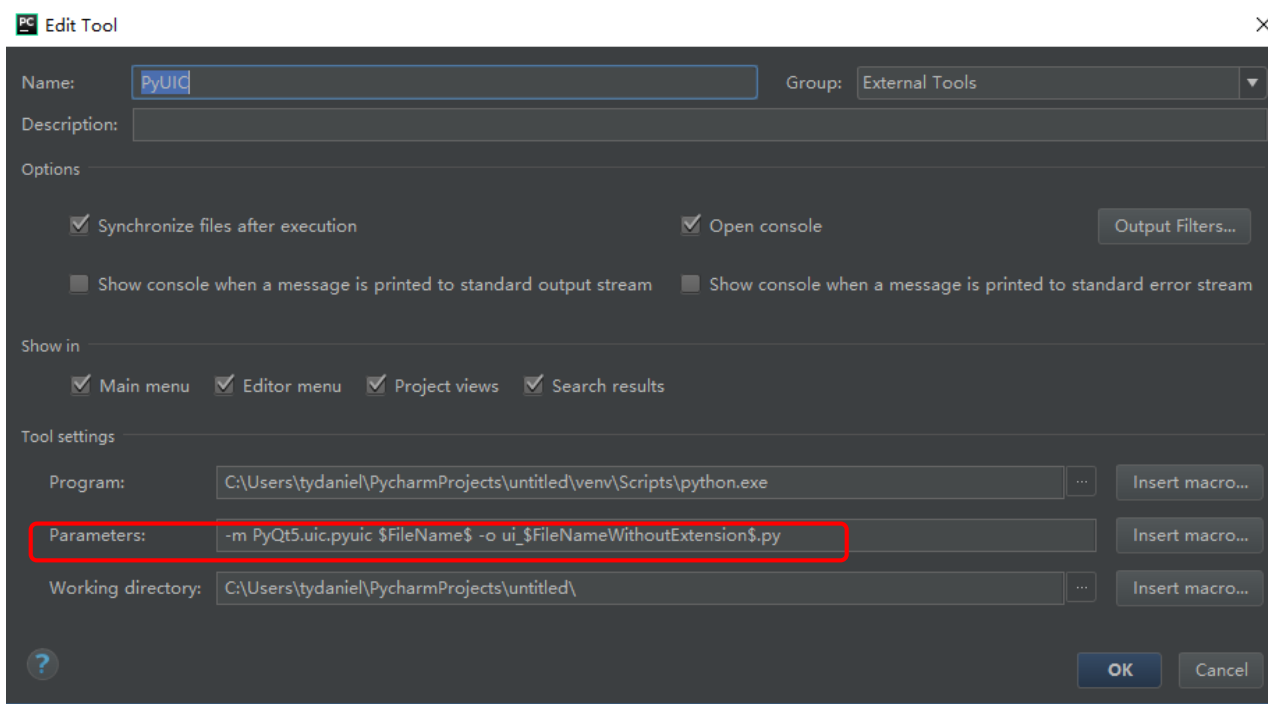
Repositories:

<https://mirrors.aliyun.com/pypi/simple/>

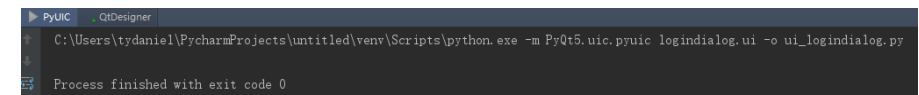
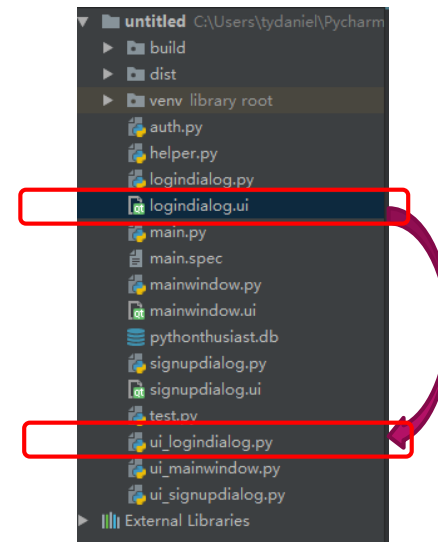
External Tools -- QtDesigner



External Tools -- PyUIC



To convert a Qt file(*.ui)
to Python script



Python Application

TTD

UNITTEST BY PYTHON

PyUnit

```
test.py x
1  import unittest
2  from PyQt5.QtCore import QApplication
3  from auth import Auth
4  import helper
5  import sys
6
7  class TestCase001(unittest.TestCase):
8      def testConnect(self):
9          qApp = QApplication(sys.argv)
10         self.assertTrue(helper.dbConnect())
11
12     def testMd5(self):
13         self.assertEqual(helper.computeHash("password"), "5f4dcc3b5aa765d61d8327deb882cf99")
14
15     def testAuth(self):
16         # self.assertEqual("actual", "expected")
17         qApp = QApplication(sys.argv)
18         helper.dbConnect()
19         auth = Auth()
20         self.assertEqual(auth.doLogin("pxj", "password"), True)
21
```

```
21
22     def testSignup(self):
23         qApp = QApplication(sys.argv)
24         helper.dbConnect()
25         auth = Auth()
26         self.assertEqual(auth.doSignup("pxj", "password"), False)
27         self.assertEqual(auth.doSignup("ipxj", "password"), True)
28
29     if __name__ == "__main__":
30         unittest.main()
```

```
pyinstaller test
C:\Users\tydaniel\PycharmProjects\untitled\venv\Scripts\python.exe C:/Users/tydaniel/PycharmProjects/untitled/test.py
....
-----
Ran 4 tests in 0.594s
OK
Process finished with exit code 0
```

Python Application

MAKE A INSTALLER FOR
PYTHON PROJECT BY
PYCHAM

PyInstaller 3.3 -- Introduction

[Docs](#) » PyInstaller Manual

[Edit on GitHub](#)

PYINSTALLER MANUAL

Version: PyInstaller 3.3.1

Homepage: <http://www.pyinstaller.org>

Contact: pyinstaller@googlegroups.com

Authors: David Cortesi, based on structure by Giovanni Bajo & William Caban, based on Gordon McMillan's manual

Copyright: This document has been placed in the public domain.

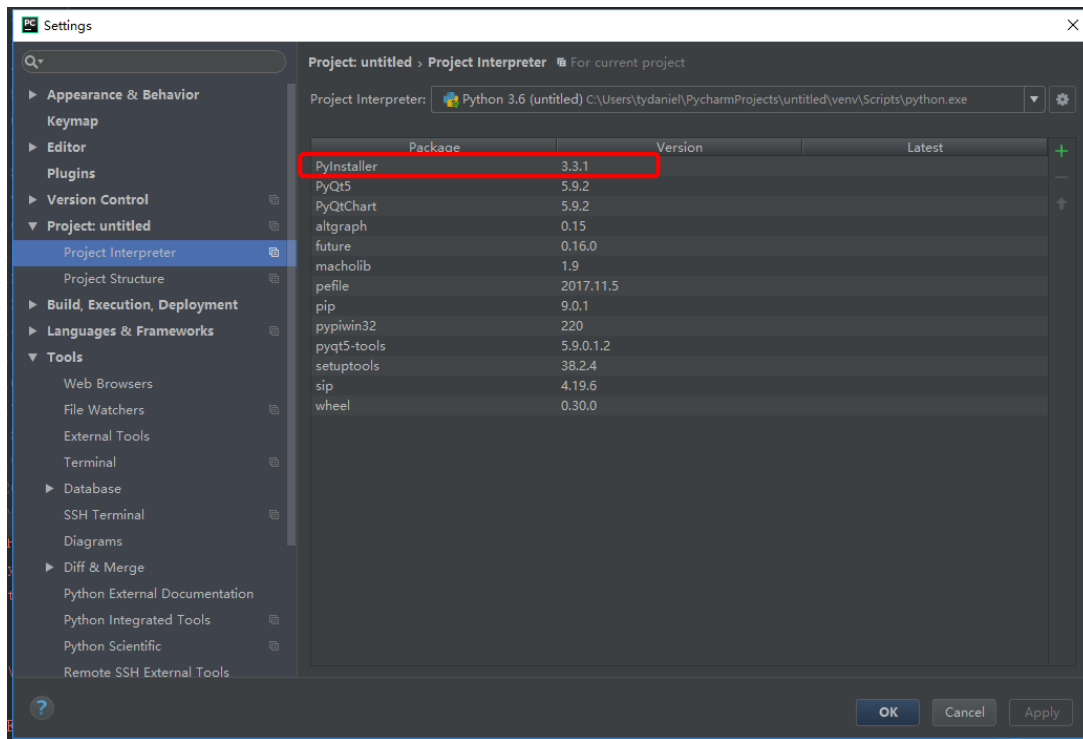
PyInstaller bundles a Python application and all its dependencies into a single package. The user can run the packaged app without installing a Python interpreter or any modules. *PyInstaller* supports Python 2.7 and Python 3.3+, and correctly bundles the major Python packages such as numpy, PyQt, Django, wxPython, and others.

PyInstaller is tested against Windows, Mac OS X, and Linux. However, it is not a cross-compiler: to make a Windows app you run *PyInstaller* in Windows; to make a Linux app you run it in Linux, etc. *PyInstaller* has been used successfully with AIX, Solaris, and FreeBSD, but is not tested against them.

Manual:

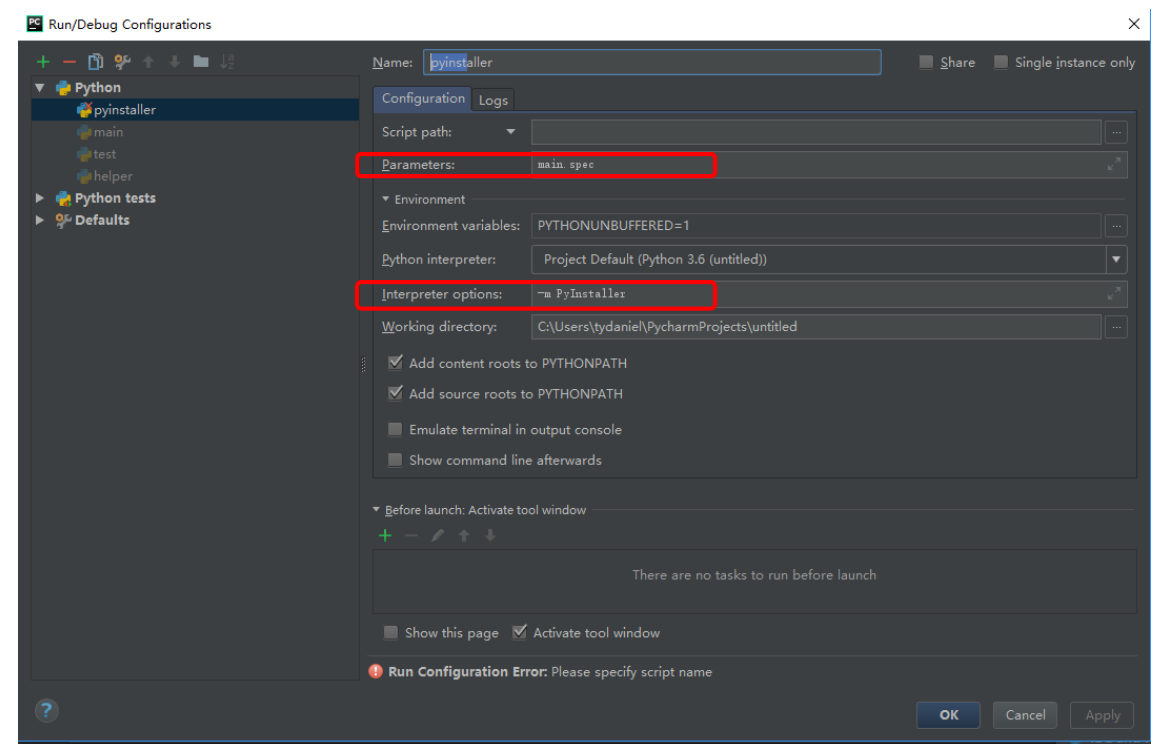
<https://pyinstaller.readthedocs.io/en/stable/index.html#>

PyInstaller 3.3 -- Preparation (Project Interpreter and Startup Tasks)



Doc Reference:

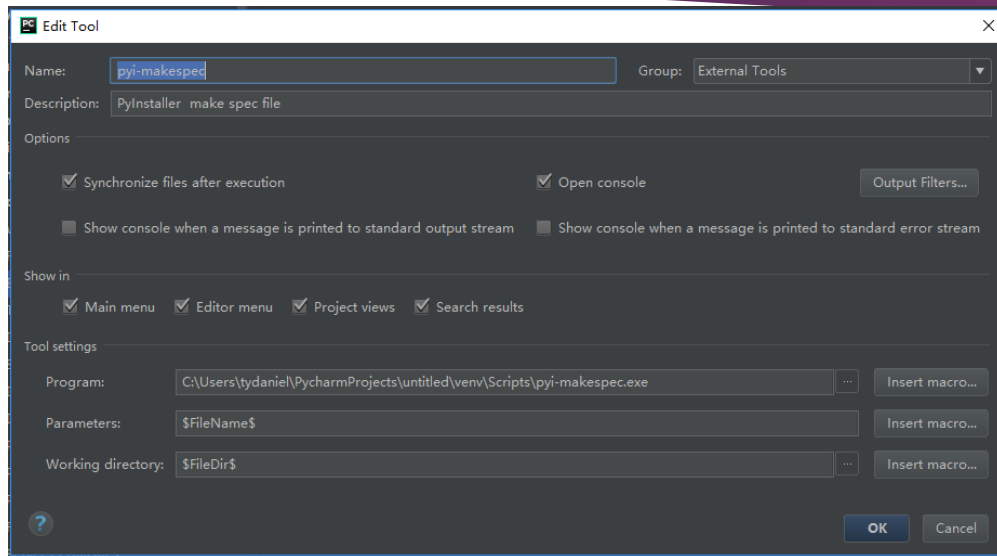
<https://pyinstaller.readthedocs.io/en/stable/>



Configure on PyCharm:

<https://stackoverflow.com/questions/33906539/configuring-pycharm-to-run-pyinstaller>

PyInstaller 3.3 – Make spec file pyi-makespec(to generate a spec file)



CLI: `pyi-makespec.exe main.py`

You create a spec file using this command:

```
pyi-makespec options name.py [other scripts ...]
```

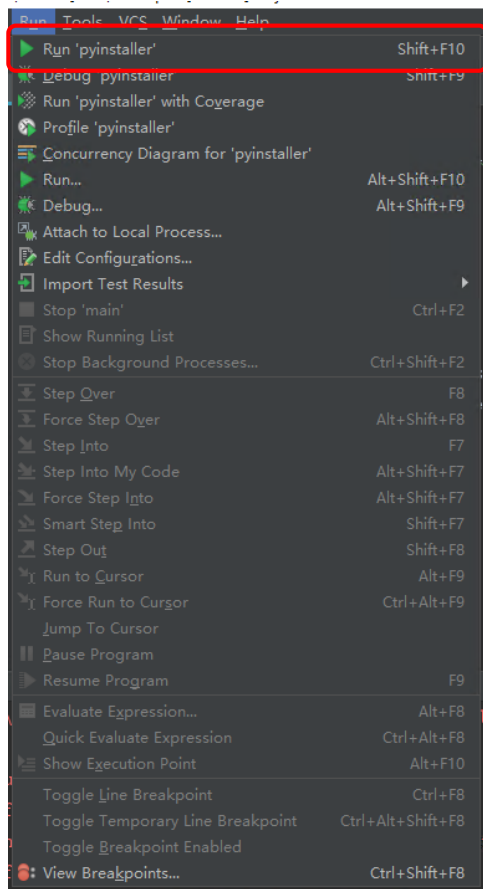
The *options* are the same options documented above for the `pyinstaller` command. This command creates the `name.spec` file but does not go on to build the executable.

```
6 a = Analysis(['main.py'],
7             pathex=['C:\\Users\\tydaniel\\PycharmProjects\\untitled'],
8             binaries=[],
9             datas=[('pythonthusiast.db', '.')],
10            hiddenimports=[],
11            hookspath=[],
12            runtime_hooks=[],
13            excludes=[],
14            win_no_prefer_redirects=False,
15            win_private_assemblies=False,
16            cipher=block_cipher)
17 pyz = PYZ(a.pure, a.zipped_data,
18          cipher=block_cipher)
19 exe = EXE(pyz,
20          a.scripts,
21          exclude_binaries=True,
22          name='main',
23          debug=False,
24          strip=False,
25          upx=True,
26          console=False)
27 coll = COLLECT(exe,
28                a.binaries,
29                a.zipfiles,
30                a.datas,
31                strip=False,
32                upx=True,
33                name='main')
```

Doc Ref:

<https://pyinstaller.readthedocs.io/en/stable/spec-files.html>

PyInstaller 3.3 -- To bundle apps and all dependences into a package(Exe file)



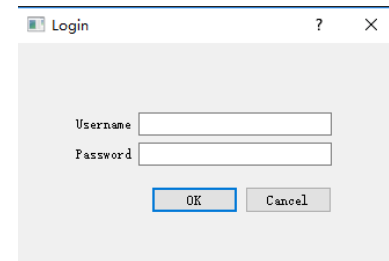
```
Run: pyinstaller -> QtDesigner
C:\Users\tydaniel\PycharmProjects\untitled\venv\Scripts\python.exe -m PyInstaller main.spec
1270 INFO: PyInstaller: 3.3.1
1274 INFO: Python: 3.6.1
1275 INFO: Platform: Windows-10-10.0.16299-SP0
1295 INFO: UPX is not available.
1296 INFO: Extending PYTHONPATH with paths
['C:\\Users\\tydaniel\\PycharmProjects\\untitled',
'C:\\Users\\tydaniel\\PycharmProjects\\untitled']
1297 INFO: checking Analysis
1400 INFO: Appending 'datas' from .spec
1433 INFO: checking PYZ
1508 INFO: checking PKG
1528 INFO: Bootloader C:\Users\tydaniel\PycharmProjects\untitled\venv\lib\site-packages\PyInstaller\bootloader\Windows-64bit\runw.exe
1528 INFO: checking EXE
1533 INFO: Building because console changed
1533 INFO: Building EXE from out00-EXE.toc
1535 INFO: Appending archive to EXE C:\Users\tydaniel\PycharmProjects\untitled\build\main\main.exe
1704 INFO: Building EXE from out00-EXE.toc completed successfully.
1713 INFO: checking COLLECT
1727 INFO: Building COLLECT out00-COLLECT.toc
8947 INFO: Building COLLECT out00-COLLECT.toc completed successfully.
Process finished with exit code 0
```

The image shows the 'dist' directory in PyCharm. A large blue arrow points from the terminal window to this directory view.

名称	修改日期	类型	大小
PyQt5	2017/12/21 16:13	文件夹	
_bz2.pyd	2017/12/21 15:03	PYD 文件	86 KB
_elementtree.pyd	2017/12/21 15:09	PYD 文件	152 KB
_hashlib.pyd	2017/12/21 15:03	PYD 文件	1,410 KB
_lzma.pyd	2017/12/21 15:04	PYD 文件	242 KB
_socket.pyd	2017/12/21 15:04	PYD 文件	64 KB
_ssl.pyd	2017/12/21 15:03	PYD 文件	1,697 KB
api-ms-win-core-console-l1-1-0.dll	2017/12/21 15:03	应用程序扩展	19 KB
api-ms-win-core-datetime-l1-1-0.dll	2017/12/21 15:03	应用程序扩展	19 KB
api-ms-win-core-debug-l1-1-0.dll	2017/12/21 15:03	应用程序扩展	19 KB
api-ms-win-core-errorhandling-l1-1-0.dll	2017/12/21 15:03	应用程序扩展	19 KB
api-ms-win-core-file-l1-1-0.dll	2017/12/21 15:03	应用程序扩展	19 KB
api-ms-win-core-file-l1-2-0.dll	2017/12/21 15:03	应用程序扩展	19 KB
api-ms-win-core-file-l2-1-0.dll	2017/12/21 15:03	应用程序扩展	19 KB
api-ms-win-core-handle-l1-1-0.dll	2017/12/21 15:03	应用程序扩展	19 KB
api-ms-win-core-heap-l1-1-0.dll	2017/12/21 15:03	应用程序扩展	19 KB
api-ms-win-core-interlocked-l1-1-0.dll	2017/12/21 15:03	应用程序扩展	19 KB
api-ms-win-core-libraryloader-l1-1-0.dll	2017/12/21 15:03	应用程序扩展	19 KB
api-ms-win-core-localization-l1-2-0.dll	2017/12/21 15:03	应用程序扩展	21 KB
api-ms-win-core-memory-l1-1-0.dll	2017/12/21 15:03	应用程序扩展	19 KB
api-ms-win-core-namedpipe-l1-1-0.dll	2017/12/21 15:03	应用程序扩展	19 KB
api-ms-win-core-processenvironment-l1-1-0.dll	2017/12/21 15:03	应用程序扩展	20 KB

The image shows the 'dist' directory in PyCharm. The file 'main.exe' is highlighted with a red rectangle.

名称	修改日期	类型	大小
api-ms-win-crt-time-l1-1-0.dll	2017/12/21 15:03	应用程序扩展	21 KB
api-ms-win-crt-utility-l1-1-0.dll	2017/12/21 15:04	应用程序扩展	19 KB
base_library.zip	2017/12/21 15:09	压缩(zip)文件...	721 KB
LIBEAY32.dll	2017/12/21 15:10	应用程序扩展	1,786 KB
LIBPQ.dll	2017/12/21 15:09	应用程序扩展	158 KB
main.exe	2017/12/21 16:13	应用程序	1,480 KB
MANIFEST-1	2017/12/21 16:13	MANIFEST 文件	2 KB
MSVCP140.dll	2017/12/21 15:09	应用程序扩展	625 KB
pyexpat.pyd	2017/12/21 15:03	PYD 文件	177 KB
PyQt5.Qt.pyd	2017/12/21 15:09	PYD 文件	12 KB
PyQt5.QtCore.pyd	2017/12/21 15:09	PYD 文件	2,180 KB
PyQt5.QtGui.pyd	2017/12/21 15:09	PYD 文件	2,293 KB
PyQt5.QtPrintSupport.pyd	2017/12/21 15:09	PYD 文件	254 KB
PyQt5.QtSql.pyd	2017/12/21 15:09	PYD 文件	299 KB
PyQt5.QtWidgets.pyd	2017/12/21 15:09	PYD 文件	4,908 KB
python3.dll	2017/12/21 15:09	应用程序扩展	50 KB
python36.dll	2017/12/21 15:03	应用程序扩展	4,028 KB
python36.db	2017/12/21 13:58	Data Base File	12 KB
Qt5Core.dll	2017/12/21 15:09	应用程序扩展	5,647 KB
Qt5Gui.dll	2017/12/21 15:09	应用程序扩展	5,910 KB
Qt5PrintSupport.dll	2017/12/21 15:09	应用程序扩展	314 KB
Qt5Sql.dll	2017/12/21 15:09	应用程序扩展	199 KB



Python Application

PYINSTALLER
ON LINUX (UBUNTU)

Three stages

- ▶ **Step1:** convert ui file into py file

Python -m PyQt5.uic.pyuic [ui file path] -o [output py file]

e.g: python -m PyQt5.uic.pyuic logindialog.ui -o ui_logindialog.py

- ▶ **Step2:** create a spec file (like make file) by pyi-makespec

Pyi-makespec [main python file]

e.g: pyi-makespec main.py => create main.spec file

- ▶ **Step3:** compile and build by PyInstaller

Python -m PyInstaller [spec file]

e.g: python -m PyInstaller main.spec

Reference

► Reference:

http://pythonthusiast.pythonblogs.com/230_pythonthusiast/categories/426_qt_pyqt_and_pyside_tutorial.html

E-Book:

Rapid GUI Programming with Python and Qt