

Debugging Blender with Pycharm

John Newton 7/7/2020

see online at [Debugging Blender with Pycharm](#)

Debugging Blender with Pycharm

Preparation (Linux and Windows)

Other debugging options and notes For Debugging under Windows:

Debugging an Installed addon:

Further notes::

Problems Connecting:

Notes:

For ease of navigation, you can also add the autocomplete path to the Project Structure

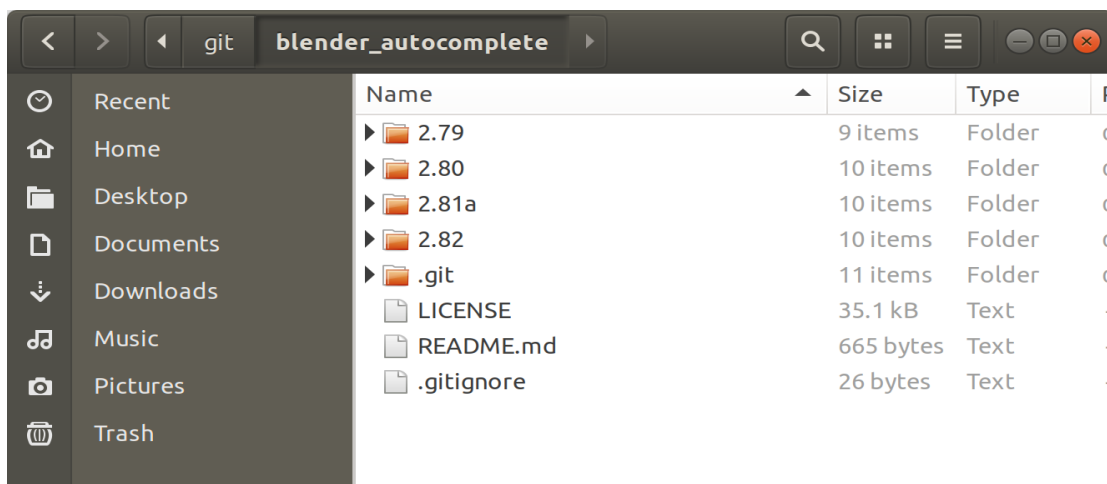
Setup the Run/Debug Configuration for remote debugging

Blender documentation

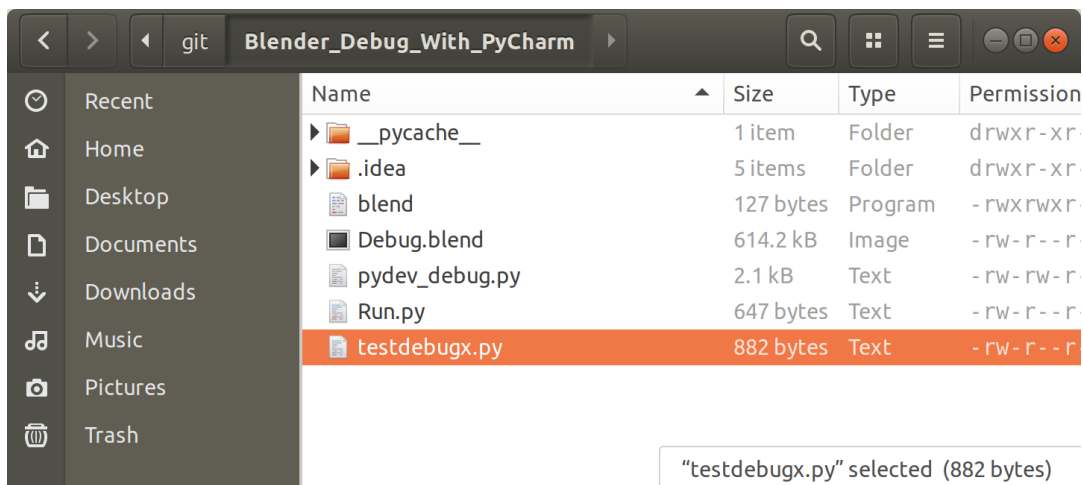
1
1
4
6
8
8
9
9
9
9

Preparation (Linux and Windows)

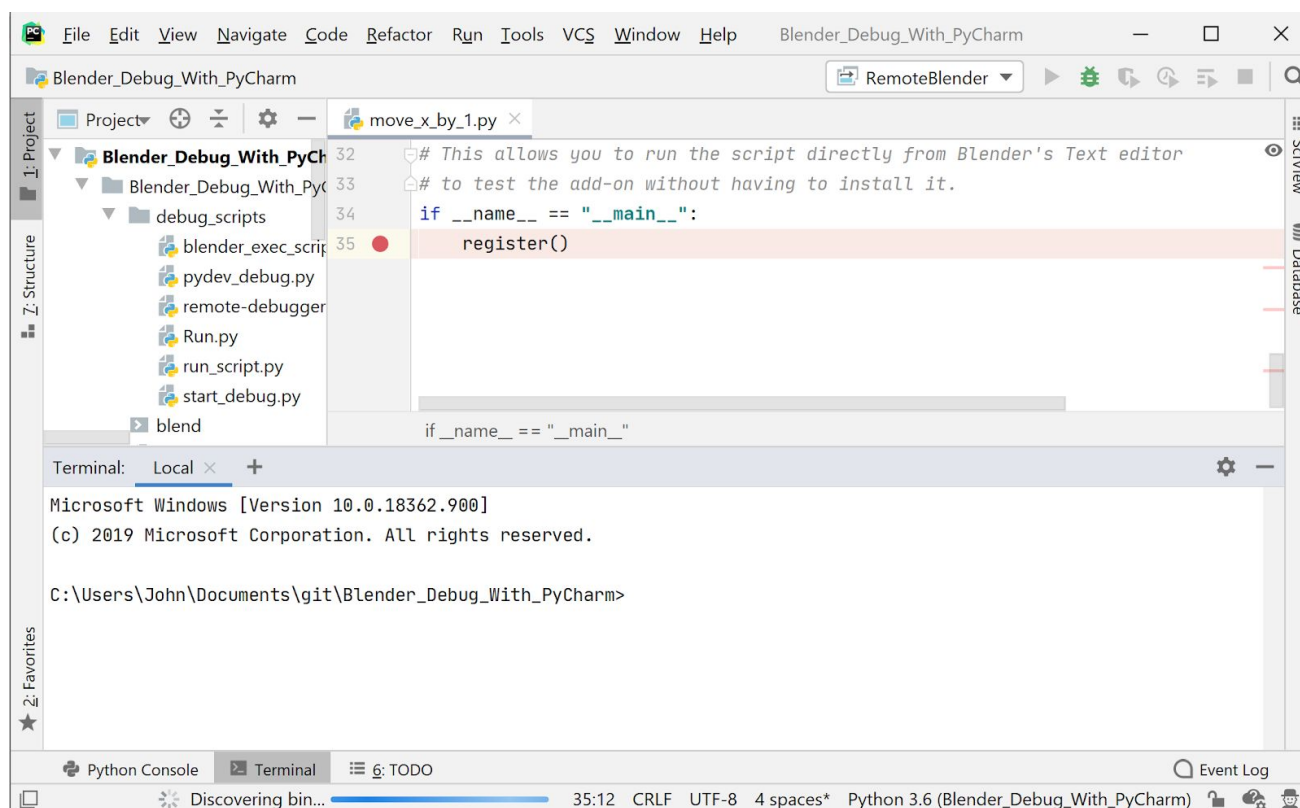
Download blender python library autocomplete from [GitHub](#) and set up in PyCharm according to [Using external IDE PyCharm for writing Blender scripts](#)



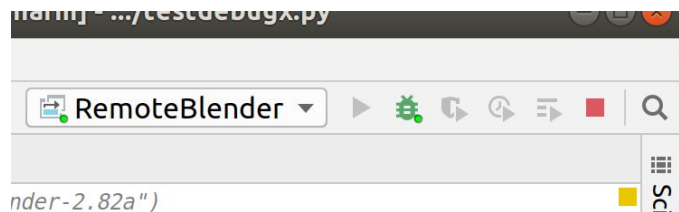
Install the test directory Files in the test directory



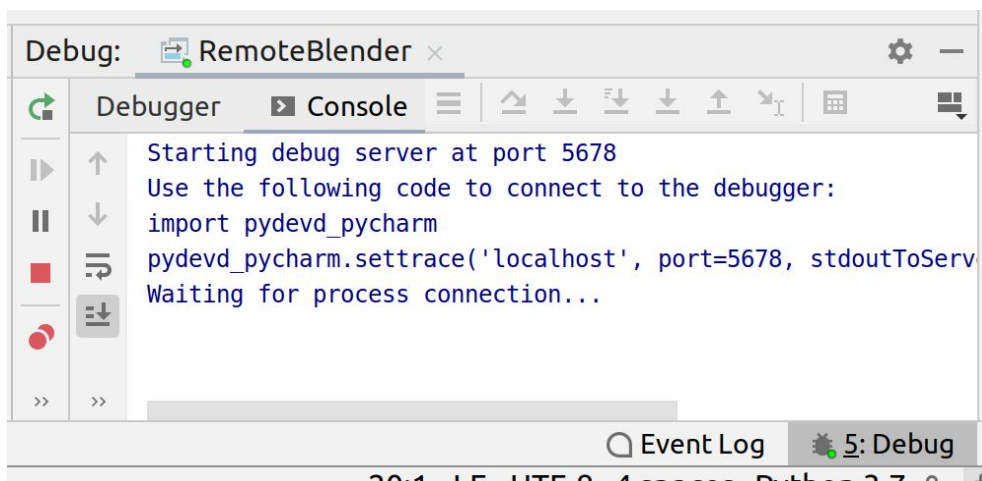
Run PyCharm and open `Blender_Debug_With_Pycharm` directory. It should load with the following configuration (at least in Linux)



To debug press the remote debug icon in the top right corner



It should respond with a prompt in the debug window ... Waiting for process connection...



In the terminal window run the **blend** bash script or in windows the **blend.bat** file. This just runs blender with the command line

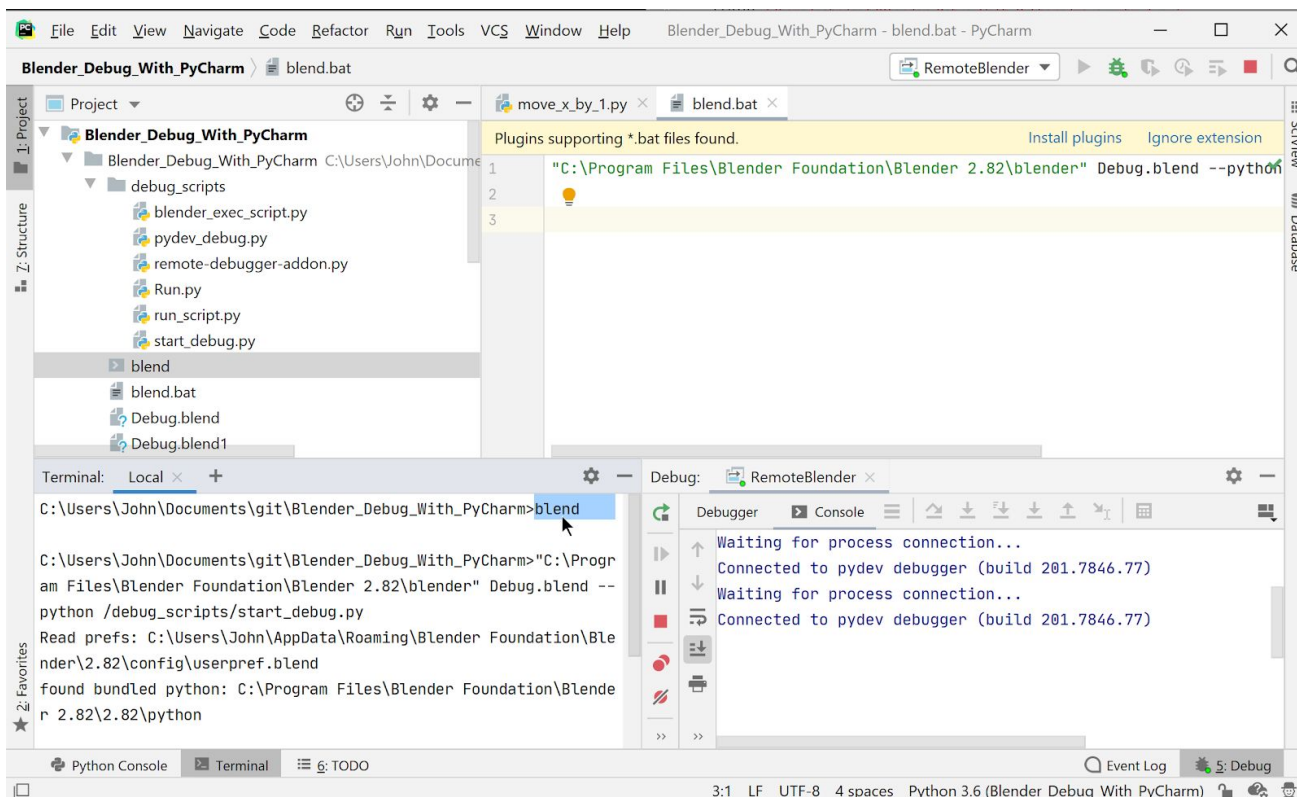
Linux: `/home/john/blender-2.82a/blender boolean.blend`

Windows: `"C:\Program Files\Blender Foundation\Blender 2.82\blender" Debug.blend`

The batch or bash file is configured to run Blender and load the Debug.blend file along with start_debug.py with the following command line

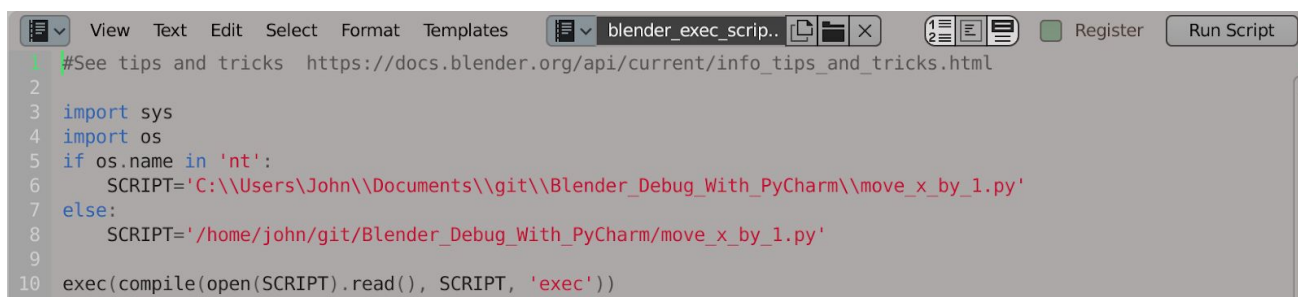
```
"C:\Program...\blender" Debug.blend --python /debug_scripts/start_debug.py
```

start_debug.py will attempt to connect to the Pycharm debugger on port 5678

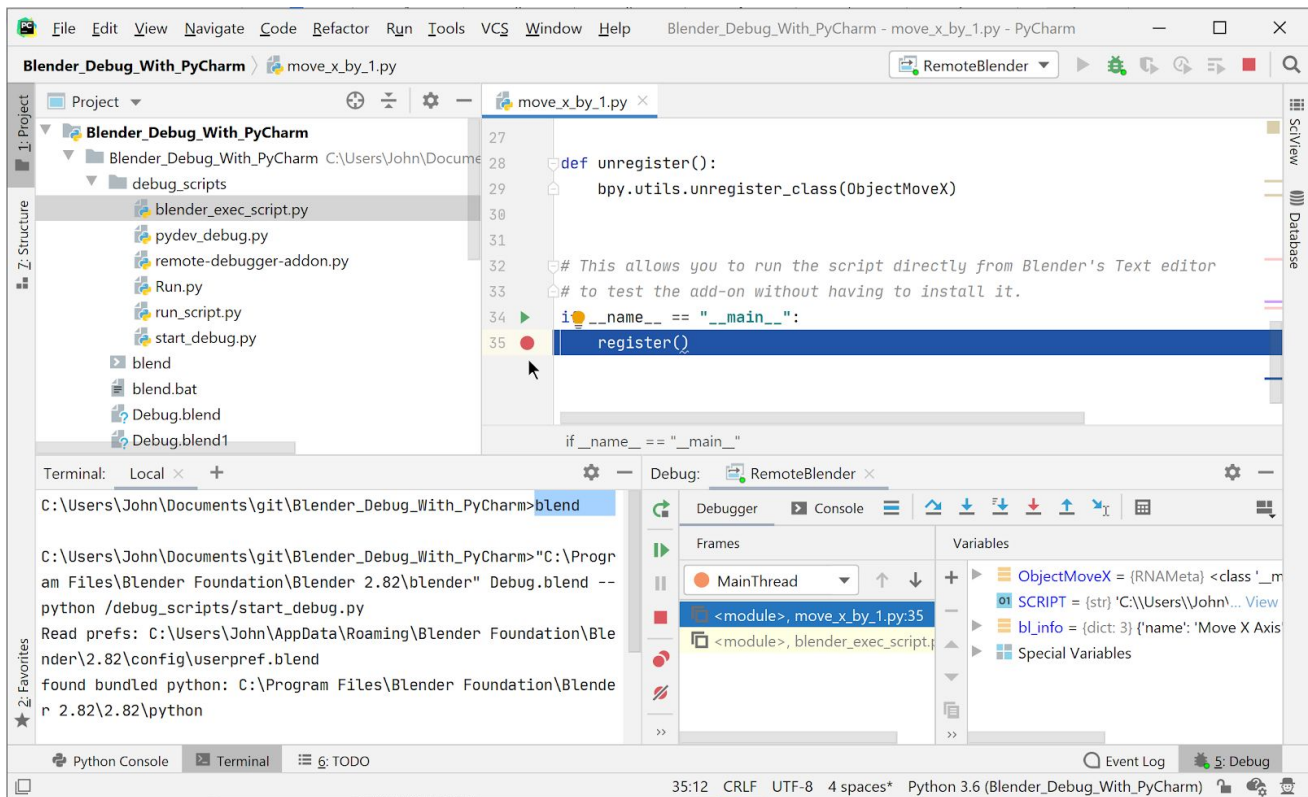


Here is a successful connection

All going well, Blender should load. In Blender navigate to the script window if it is not showing. If the **blender_exec_script.py** script is not visible navigate to it. You will need to set this up to point to the required script file that you want to execute too. See [Tips and Tricks — Blender Python API](#) for other calling methods.



Press the **Run Script** button in the top right corner. The Pycharm debugger should connect and it should run to the breakpoint on line 35. The bpy_struct object should be shown as a variable. You can now single-step through the rest of the program.



Other debugging options and notes For Debugging under Windows:

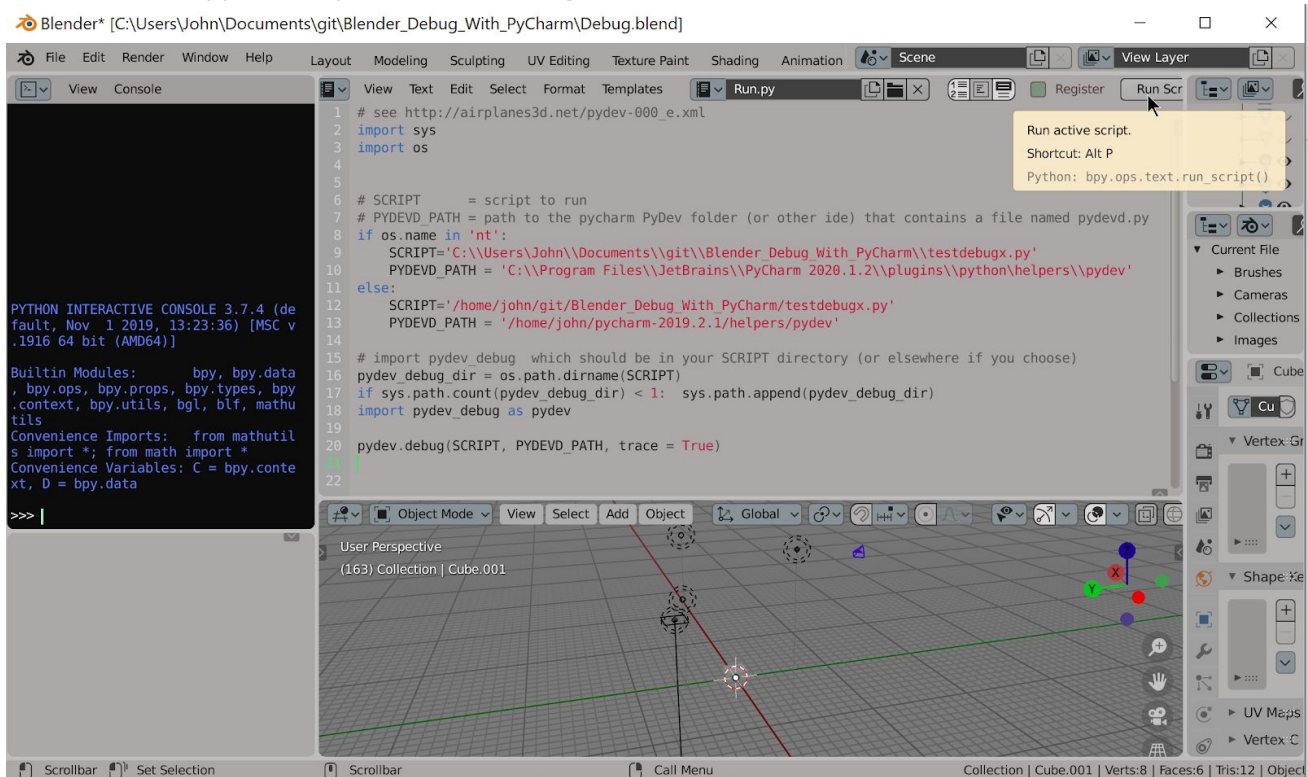
There are other techniques that can be used for instance see:

[Programming Add-ons for Blender 2.8](#) by Witold Jaworski

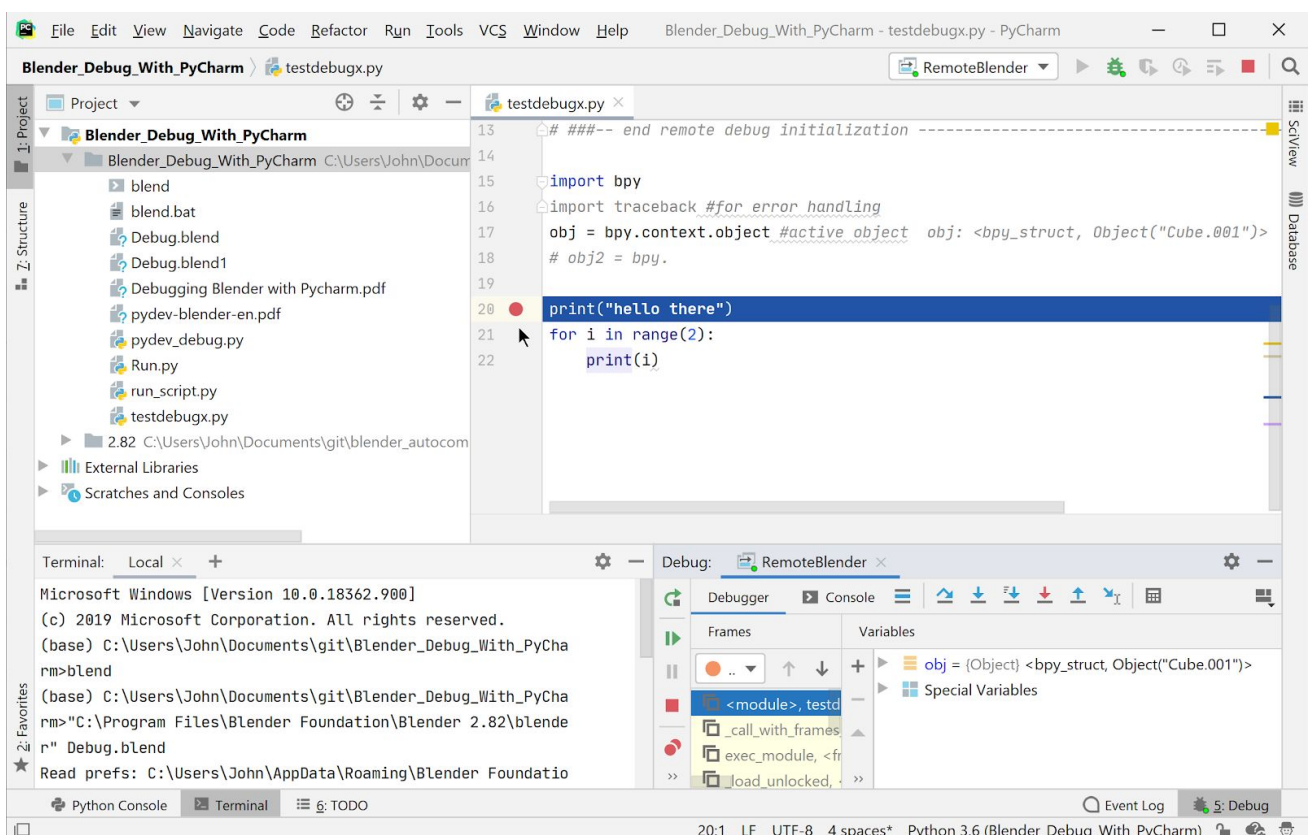
[sybrenstuvell / random-blender-addons / remote_debugger.py](#) on Github

Here is shown **Witold Jaworski's** method which connects to the debugger in the main script. This may be useful in some situations.

In the blender application press the **Run Script** button



The debugger should connect and stop at the **breakpoint** here shown at line 20



It is then possible to step through lines.... The blender application will be unresponsive while the python code is stopped at a breakpoint. It can only run one python script at one time and the current script will block all others.

Debugging an Installed addon:

In windows, the Blender addon directory is at

Windows: C:\Users\<USER>\AppData\Roaming\Blender Foundation\Blender\2.82\scripts\addons

Linux: /home/...

This is where the zip files are extracted to. I have added a batch file and a debug.blend to make allow Blender to conveniently run from PyCharm.

Blender Foundation > Blender > 2.82 > scripts > addons >				Search add...
<input type="checkbox"/>	Name	Date modified	Type	Size
<input type="checkbox"/>	.idea	6/07/2020 6:38 PM	File folder	
<input type="checkbox"/>	camera-image-project-master	14/06/2020 9:50 PM	File folder	
<input type="checkbox"/>	Lumiere	6/07/2020 6:28 PM	File folder	
<input type="checkbox"/>	projectile	6/07/2020 6:23 PM	File folder	
<input checked="" type="checkbox"/>	blend.bat	6/07/2020 12:59 PM	Windows Batch File	1 KB
<input checked="" type="checkbox"/>	Debug.blend	6/07/2020 6:29 PM	Blender File	596 KB

In the addon `__init__.py` file add the following code block near the top of the file

```
##### DEBUG CODE #####
import sys
import os

# PYDEV_PATH = path to the pycharm PyDev folder (or other ide) that contains a file named pydevd.py
if os.name in 'nt':
    PYDEV_PATH = 'C:\\Program Files\\JetBrains\\PyCharm 2020.1.2\\plugins\\python\\helpers\\pydev'
else:
    PYDEV_PATH = '/home/<USER>/pycharm-2019.2.1/helpers/pydev'

if sys.path.count(PYDEV_PATH) < 1: sys.path.append(PYDEV_PATH)
import pydevd_pycharm
pydevd_pycharm.settrace('localhost', port=5678, stdoutToServer=True, stderrToServer=True,
suspend=False)
#####
```

Running the blend.bat file from the terminal will then allow a connection to the debugger in Pycharm.

As before we have set up the [autocomplete path to the Project Structure](#) and also [Setup the Run/Debug Configuration for remote debugging](#).

The figure below shows where a breakpoint has been added to the **register()** function.

Breakpoints can be set to any file or procedure that is called in the session.

Blender addons - _init_.py - PyCharm

addons > Lumiere > _init_.py

Project Structure: addons > Lumiere > _init_.py

```
58 #===== DEBUG CODE =====
59 import sys
60 import os
61
62 # PYDEV_PATH = path to the pycharm PyDev folder (or other ide) that contains a file named
63 if os.name in 'nt':
64     # SCRIPT = 'C:\\Users\\John\\Documents\\git\\Blender_Debug_With_PyCharm\\testdebugx.py'
65     PYDEV_PATH = 'C:\\Program Files\\JetBrains\\PyCharm 2020.1.2\\plugins\\python\\helpers\\pydev'
66 else:
67     # SCRIPT = '/home/john/git/Blender_Debug_With_PyCharm/testdebugx.py'
68     PYDEV_PATH = '/home/john/pycharm-2019.2.1/helpers/pydev'
69
70 if sys.path.count(PYDEV_PATH) < 1: sys.path.append(PYDEV_PATH)
71 import pydevd_pycharm
72 pydevd_pycharm.settrace('localhost', port=5678, stdoutToServer=True, stderrToServer=True)
73 #=====
74
75 # register
76 def register():
77     lumiere_gizmo.register()
78     lumiere_op.register()
79
80 register()
```

Terminal: Local × +

```
2\\scripts\\addons>bblend
C:\\Users\\John\\AppData\\Roaming\\Blender Foundation\\Blender\\2.8
2\\scripts\\addons>"C:\\Program Files\\Blender Foundation\\Blender\\2.82\\blender" Debug.blend
Read prefs: C:\\Users\\John\\AppData\\Roaming\\Blender Foundation
```

Debug: Blender Debug ×

Debugger: MainThread, register, _init_.py:77, enable, addon_utils.py:382

Variables: Variables are not available

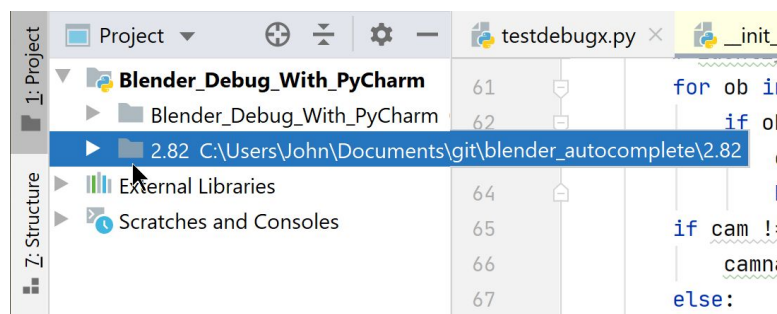
Event Log: 2

Windows Defender configuration updated (48 minutes ago)

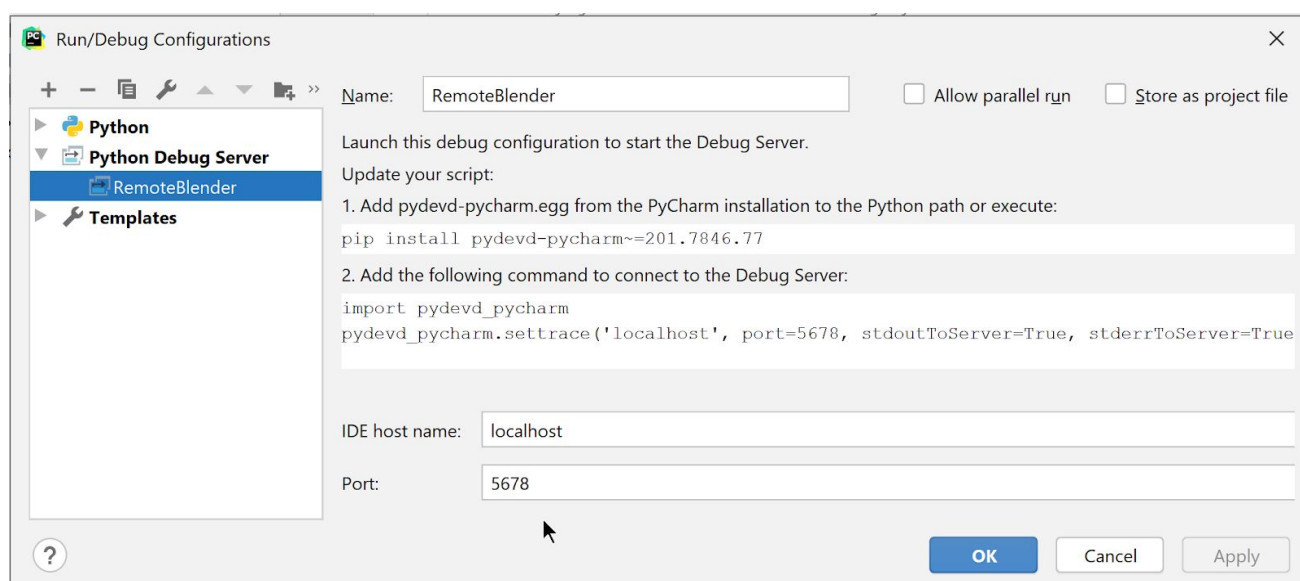
77:1 CRLF UTF-8 Tab Python 3.7 (Blender_Debug_With_PyCharm)

Further notes::

[For ease of navigation, you can also add the autocomplete path to the Project Structure](#)

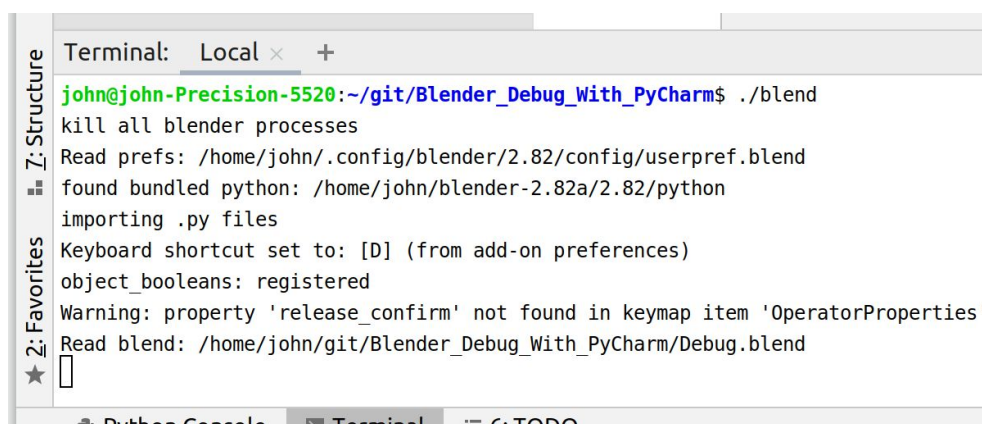


[Setup the Run/Debug Configuration for remote debugging](#)



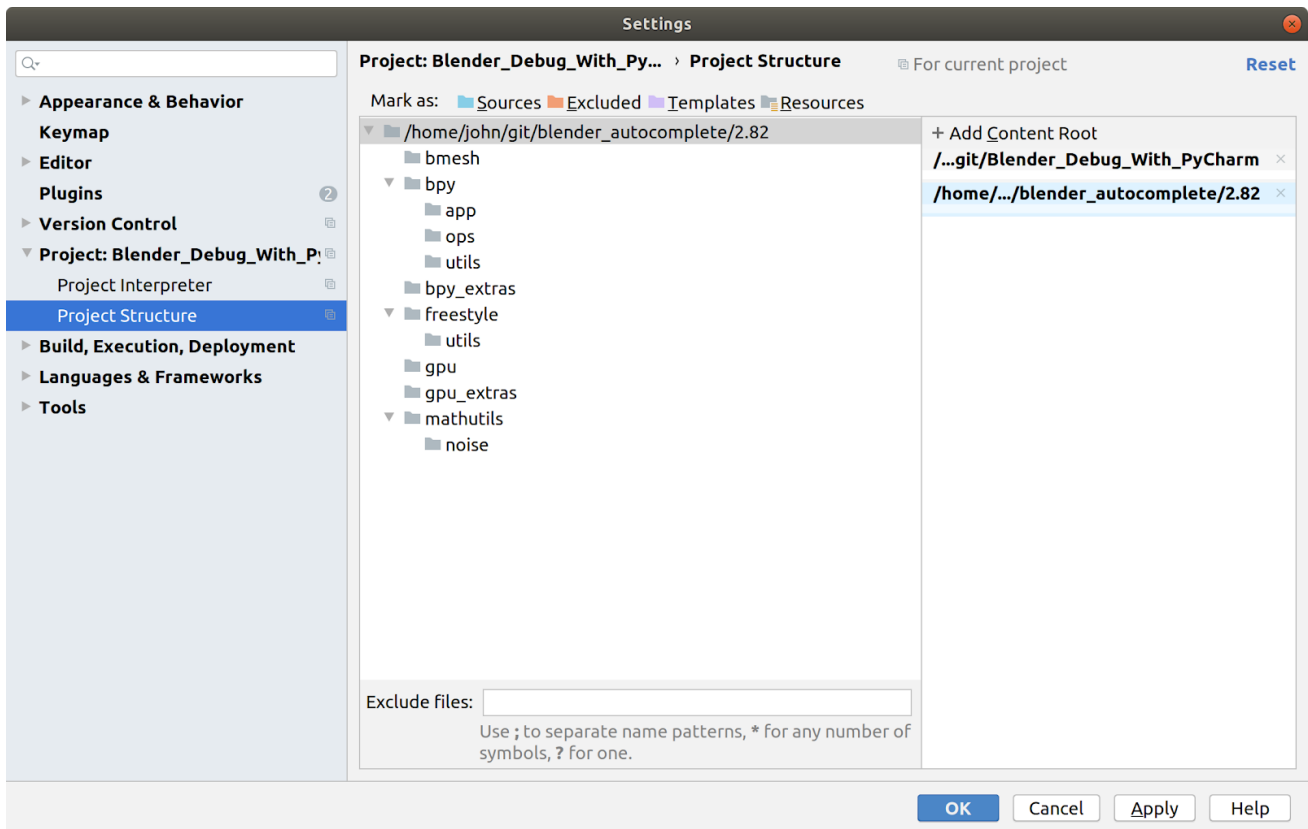
Problems Connecting:

- Check all the directories
- Close the local terminal and restart blender with **./blend**, the Linux script will kill all existing blender processes

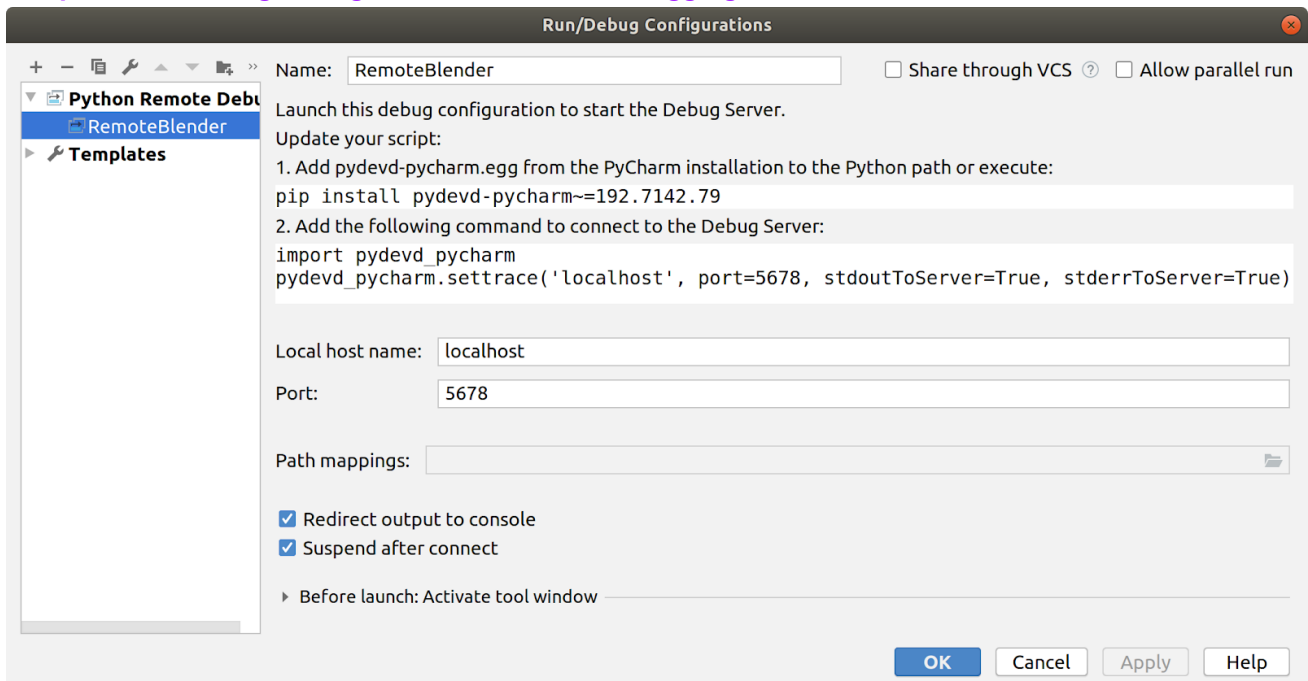


Notes:

For ease of navigation, you can also add the autocomplete path to the Project Structure



Setup the Run/Debug Configuration for remote debugging



Blender documentation

[Blender 2.83.0 Python API Documentation — Blender Python API](#)

https://docs.blender.org/api/current/info_tips_and_tricks.html

