Graphical user interface, application

Description automatically generated

DRAFT

This is an instruction file to run python scripts from Excel using xlwings library. The Location\_Coordinates.xlsx file is used in this example, with the Location\_Coordinates.py script. This example will create an output.xlsx spreadsheet file with the appropriate p-code to the geographic coordinates listed in the input spreadsheet columns. Before using xlwings, make sure that python is installed in your system. You can check it by typing “**python --version**” on command prompt and you should get the available version in return.

Text

Description automatically generated

If python isn’t available in your system, you can directly download it from [here](https://www.python.org/downloads/) or you can follow this easy instructions available [here](https://www.ics.uci.edu/~pattis/common/handouts/pythoneclipsejava/python.html).

Once you are done with installing python, you might need to install several packages which didn’t come by default with python. To install any package, use this command in cmd: pip install <package name>. Example: pip install pandas.

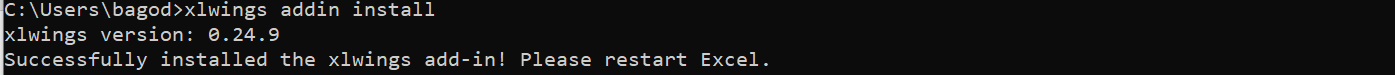
Now, follow these steps to enable python script with Excel.

**Step 1: Install xlwings library**

Go to command prompt (cmd) and type this command: **pip install xlwings.**

**Step 2: Install Excel addin**

Once the library has been installed, we need to install an addin for Excel. Ensure you have closed all your Excel instance before running this command on cmd: **xlwings addin install**. The successful installation of the addin is shown in the below picture



**Step 3: Confirming the addin on Excel**

Your Excel will show this new ribbon named “xlwings”

Graphical user interface, application, table, Excel

Description automatically generated

**Step 4: Enabling User Defined Functions for xlwings**

To enable this function, press Alt+L+H to get this window and check the xlwings option.

Graphical user interface, application

Description automatically generated

If xlwings isn’t showing in your addin options then you have to download the addin from this link [here](https://github.com/xlwings/xlwings/releases/download/0.24.9/xlwings.xlam) and save it to this path C:\Users\**xxxx**\AppData\Roaming\Microsoft\AddIns

Where xxxx is your username in your computer. Now click on browse option which will automatically take you to the Microsoft\AddIns folder, if not, you can go there manually using this path C:\Users\**xxxx**\AppData\Roaming\Microsoft\AddIns

Select the xlwings.xlam file you just downloaded.

**Step 5: Enable Trust Access**

Finally, you need to enable the trust access to the VBA project. You can do this by navigating to File > Options> Trust Center > Trust Center Setting > Macro Setting

Graphical user interface, text, application

Description automatically generated

**Step 6: Setting Interpreter path (optional)**

Click on xlwings ribbon and check if it has defaultly filled the interpreter path or not

Graphical user interface, text

Description automatically generated

If the interpreter path isn’t filled, then you need to find the python interpreter’s path with this command on cmd: ‘where python’. Paste the path to the interpreter section of xlwings. Finally, check the show console option available at the ribbon. Additionally make sure the “Show Console” checkbox is checked in the “Advanced” column of the ribbon.

**Step 7: Click on the Run main button**

When you click on the run main button, it will invoke the main method of python script and command prompt will open. Make sure that the excel file and python script should have same name (Location\_Coordinates in our case). Follow the command prompt instructions by typing in the window “y” to continue or “n” to exit. The processing will continue if so directed, and the script may take some time to run. In the same directory where the spreadsheet exists, a log file will be created and the console will indicate when the process is complete and the output file is accessible.