Setting up the WF TDM Python Environment

Step 1: Download WF TDM Python Environment

The WF TDM Python Environment (py-tdm-env) can be downloaded here. Extract the contents of the zipped folder to a strategic location on your computer. We suggest extracting to the following location: `C:\Python\py-tdm-env`. You may need to create your own folder named `Python`. If you do not permission to extract to the suggested location, extract somewhere else that makes sense.

Step 2: Open Visual Studio Code

If you don't have Visual Studio Code, download it here. Next, open the `py-tdm-env` folder in Visual Studio. This can be done by right clicking on the folder in File Explorer, and selecting `Open with Code`. Alternatively, you can open Visual Studio and select `File/Open Folder` from the menu selection items and navigate to the folder.

You can use a different text editor, however we recommend the Visual Studio IDE platform.

Step 3: Download Python and Jupyter Extensions

After open Visual Studio, ensure that the `Python` and `Jupyter` extensions are downloaded. You can do this by navigating to `Extensions` window or pressing `Ctrl+Shift+X`. Next search for `Python` and `Jupyter` extensions and download them.

Step 4: Refresh Visual Studio

- Press `Ctrl+Shift+P` to open the command prompt in Visual Studio.
- Type `Reload: Window` and select it.
- After reloading, close out Visual Studio and then reopen it.
- Open or create a Jupyter Notebook file (`.ipynb`).

Step 5: Locate the Python Interpreter

- Press `Ctrl+Shift+P`
- Type `Python: Select Interpreter`
- Navigate to the `python.exe` path that exists within the python environment: `C:\Python\py-tdm-env`.

Step 6: Create a New Kernel Spec

Open a terminal or command prompt window. In Visual Studio, this can be done by selecting `Terminal/New Terminal` from the menu selection items. In the terminal window copy, paste, and run the following code to create a new kernel specification with the python interpreter.

```
# Install kernelspec for future use of jupyter notebooks
C:/Python/py-tdm-env/python.exe -m ipykernel install --user --name=py-tdm-env --
display-name "py-tdm-env"
```

After creating the new kernel spec, reload Visual Studio (*See Step 4*).

Note: The purpose of this command is to install a new Jupyter kernel for the Python environment located in the current directory. It will install the kernel for the current user only and in a user specific directory. You do not need to install the `ipykernel` library because it already exists in the Python environment.

Step 7: Select py-tdm-env Kernel

The new environment should be automatically selected. If its not automatically selected, select the newly created kernel from the kernel selection box (top-right corner) with `Select Kernel/Jupyter Kernel../py-tdm-env`. Alternatively, you may also select the Python environment from the kernel selection box (top-right corner) with `Select Kernel/Python Environments../py-tdm-env`. Both should allow you to run cells in a Jupyter Notebook file.

In general, whenever you run a Jupyter Notebook file (.ipynb) you will need to select a kernel from the `Select Kernel` menu option. Fortunately, once you add a kernel spec that kernel is stored in the picklist for future uses. Therefore, whenever you open or reopen a .ipynb in Visual Studio, the `py-tdm-env` should always be available to select. This also means you are able to download different Python environments and add additional kernel specs to the picklist.

Note: The `Select Kernel` option only shows when a Jupyter notebook file is open.

Troubleshooting

A few basic troubleshooting ideas include:

- Reloading the Visual Studio Window
- Closing out of Visual Studio
- Restarting your Computer
- Double checking the Python environment path location
- Verifying the installation of the kernel

You can verify that the kernel spec was successfully installed by running the following command. The result should display `py-tdm-env`. If the kernel name does not show up, Step 6 was probably completed incorrectly.

```
C:/Python/py-tdm-env/jupyter.exe kernelspec list
```

If you are still having trouble, feel free to reach out to Chris Day at WFRC (cday@wfrc.org) and he'd be happy to assist.