**PyOrator setup guidance**

This guidance applies to the Windows environment only, however, since Python is available on all mainstream platforms, notably Linux and macOS, PyOrator can also be ported to these on request.

PyOrator is being developed using Python version 3.8 from the Python Software Foundation ([PSF](https://www.python.org/)) with the following additional modules installed using the Python package installer [pip](https://pypi.org/project/pip/):

* openpyxl
* pandas
* pyqt5

**An important note on the Windows environment and JSON files**

PyOrator reads and writes various setup, configuration and management files which are formatted using the JavaScript Object Notation (JSON) data interchange format.

JSON files are "self describing" and easy to understand. A JSON file uses human-readable text to store and transmit data objects consisting of attribute-value pairs and array data types.

It is highly recommended that you use Notepad++ with the JSTool plugin for editing JSON files.

Notepad++ is a text and source code editor for use with Microsoft Windows. It supports tabbed editing, which allows working with multiple open files in a single window.

Notepad++ can be downloaded here: <https://notepad-plus-plus.org/>

**Setting up PyOrator for the first time**

PyOrator is stored on a GitHub repository.

Navigate to your preferred location on your workstation and open a command prompt.

It is OK to clone PyOrator from the top level of your file system thus:

C:\> git clone <https://github.com/NewEconomicPolicy/testPyOra>

It is important to choose a staging area i.e. a location on your file system, which is separate from the program modules then copy **testPyOra\OratorRun\Docs\PyOrator model files.zip** to the staging area and unzip to that location.

**Folders created**

|  |  |
| --- | --- |
| config | Contains a model configuration file **pyorator\_config.json** |
| images | Image files for program and desktop shortcut icon |
| inputs | Contains a model ORATOR inputs Excel file |
| log | Folder for log files - not currently used |
| management | Contains a choice of model management json files |
| run | Contains a model, **pyorator.bat**, to run PyOrator |
| setup | Contains a model setup file, **pyorator\_setup.json** |

**Description of required files**

PyOrator can be run directly using a Windows desktop icon to reference a batch script file named **pyorator.bat** in the distribution**.** The batch file invokes the Python interpreter to run PyOrator program.

The desktop shortcut can be displayed with the PyOrator icon file.

**pyorator.bat:**

It is necessary to change the highlighted paths to correspond to the local filesystem

|  |  |
| --- | --- |
| Line |  |
| 1 | echo off |
| 2 | rem @ tells the command processor to be less verbose |
| 3 | @set PYTHONPATH=C:\testPyOra\BioModels;C:\testPyOra\EnvModelModules |
| 4 | @set initial\_working\_dir=%cd% |
| 5 | @chdir /D C:\ORATOR\setup |
| 6 | @C:\Python38\python.exe -W ignore C:\testPyOra\InitInptsRslts\PyOratorGUI.py |
| 7 | @chdir /D %initial\_working\_dir% |

|  |  |
| --- | --- |
| Line | Additional line description |
| 3 | extend the interpreter module search path to pick up all modules |
| 5 | change the working directory so PyOrator can locate the setup file – see below for setup file details |
| 6 | invoke the Python interpreter to read the PyOrator entry script, PyOratorGUI.py with the warning messages switched off (-W ignore) |

**The setup and configuration files**

**The setup file**

At start up PyOrator reads the setup file **pyorator\_setup.json** in the current working directory, i.e. the directory in which the program is initialised.

The setup file should be edited to point to readable paths similar to the example below.

{

"setup": {

"config\_dir": "E:\\ORATOR\\config",

"fname\_png": "E:\\ORATOR\\Images\\orator\_logo.png",

"fname\_lookup": "G:\\testPyOra\\OratorRun\\lookup\\Orator variables lookup table.xlsx",

"log\_dir": "E:\\ORATOR\\logs"

}

}

|  |  |
| --- | --- |
| config\_dir | the configuration file comprising user settings |
| fname\_png | the logo file which appears in the LH side of the user interface (GUI) |
| fname\_lookup | the single sheet Excel file comprising a lookup table for the Orator variables |
| log\_dir | the path where PyOrator log files will be written |

**The configuration file**

The PyOrator configuration file is read at program start up and overwritten with new user settings on program exit or when saved.

{

"inp\_xls": "E:\\ORATOR\\inputs\\ORATOR\_inputs.xlsx",

" mgmt\_dir": "E:\\ORATOR\\management\\Gondar",

"out\_dir": "F:\\Orator",

"use\_json": true,

"write\_excel": true

}

|  |  |
| --- | --- |
| inp\_xls | the Excel file of inputs required to run PyOrator |
| mgmt\_dir | the file location with JSON files comprising management data for steady state and forward runs |
| out\_dir | the single sheet Excel file comprising a lookup table for the Orator variables |
| use\_json | boolean value which if true makes the program read JSON files for management instead of the Inputs3b and Inputs3d sheets in Excel inputs file |
| write\_excel | boolean value file path for Excel outputs files |

**ORATOR\_inputs.xlsx**

PyOrator requires an Excel file of inputs which is based on a subset of sheets from the ORATOR Excel workbook:

Sheets required:

|  |  |
| --- | --- |
| Inputs1- Farm location |  |
| Inputs3b- Soils & Rotations | steady state management |
| Inputs3d- Changes in rotations | forward run management |
| N constants | Nitrogen constants |
| Crop parms | Crop parameters |
| Org Waste parms | Org Waste parameters |
| Weather |  |