**Python**

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

netcdf4

openpyxl

pandas

pyqt5

pyshp

PyOrator is written using the PEP 8 [Style Guide for Python](https://www.python.org/dev/peps/pep-0008/)

However, some code stanzas may not yet conform to PEP 8

**Description of required files**

**pyorator.bat:**

This Windows batch file runs PyOrator by invoking the Python interpreter, 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=E:\AbUniv\GlblEcosseModules2 |
| 4 | @set initial\_working\_dir=%cd% |
| 5 | @chdir /D E:\ORATOR\setup |
| 6 | @E:\Python38\python.exe -W ignore E:\AbUniv\PyOrator\PyOratorGUI.py |
| 7 | @chdir /D %initial\_working\_dir% |

|  |  |
| --- | --- |
| Line | Additional line description |
| 3 | extend the interpreter module search path to pick up GlobalEcosse modules |
| 4 | change the working directory so PyOrator can locate the setup file – see below for setup file details |
| 5 | invoke the Python interpreter to read the PyOrator entry script, PyOratorGUI.py with the warning messages switched off (-W ignore) |

A desktop shortcut displayed with the PyOrator icon file should be created pointing to the **pyorator.bat** file.

**The setup and configuration files**

Where possible JSON ([JavaScript Object Notation](https://www.w3schools.com/whatis/whatis_json.asp)) files are used as these 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.

**pyorator\_setup.json**

The setup file needs to be edited to point to readable paths.

Note: if the HWSD and weather sets are not present then PyOrator will use only the input Excel file for the climate values and soil parameters.

{

"setup": {

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

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

"hwsd\_dir": "G:\\GlobalEcosseData\\HWSD\_NEW",

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

"shp\_dir": "E:\\GlobalEcosseData\\CountryShapefiles",

"weather\_dir": "E:\\GlobalEcosseData\\"

}

}

|  |  |
| --- | --- |
| config\_dir | path for configuration files. The configuration file is read at program start up and overwritten with new user settings on exit |
| fname\_png | logo file which appears in the LH side of the user interface (GUI) |
| hwsd\_dir | path to Harmonized World Soil Database |
| log\_dir | path to where log files will be written |
| shp\_dir | Not active |
| weather\_dir | path to weather datasets |

**pyorator\_Haramaya.json**

A PyOrator configuration file uses an underscore character as a separator to indicate the study name which is the last segment e.g. “Haramaya” in the above example.

Many of the lines in the configuration file, see below, relate to Global Ecosse e.g. the historic and future start and end years.

{

"cmnGUI": {

"climScnr": "A1B",

"futEndYr": "2100",

"futStrtYr": "1970",

"histEndYr": "2000",

"histStrtYr": "1980",

"study": "Haramaya"

},

"minGUI": {

"aveWthrFlag": false,

"location": [

38.2,

7.5

],

"use\_xlsoil": false,

"use\_xlwthr": false,

"weatherResource": "CRU",

"xls\_inp\_fname": "E:/ORATOR/ORATOR\_inputs.xlsx"

}

}

xls\_inp\_fname identifies the Excel file of inputs required to run PyOrator.

**ORATOR\_inputs.xlsx**

A conformant Excel file of inputs is required to run PyOrator

The the Excel file is read via the **read\_input\_excel** class

**Inputs**

Sheets read:

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

**Variable naming convention used in PyOrator**

Important suffixes and prefixes:

|  |  |
| --- | --- |
| **prefix** | **meaning** |
| \_adjust \_adj\_ | adjusted |
| appld | applied |
| atmos\_ | atmospheric |
| c\_ | carbon |
| \_coef | coefficient |
| cow\_ | input of carbon from organic waste |
| cumul | cumulative |
| \_dem | demand |
| \_denit | denitrify |
| dpth\_ | depth |
| ex\_ | extra as in extra organic waste |
| factr | factor |
| \_fc | field capacity |
| fert | fertiliser |
| fwd | forward |
| grow\_dds gdds | growing degree days |
| harv\_ | harvest |
| indx | index |
| inorg | inorganic |
| inrt | inert |
| mgmt | management |
| n2 | nitrogen |
| n2o | nitrous oxide |
|  |  |
| \_nitrif | nitrification |
| \_no | nitric oxide |
| no3 | nitrogen trioxide or nitrate radical |
| nut\_ | nutriment |
| opt | optimum |
| ow\_ | organic waste |
| org\_ inorg\_ | organic inorganic |
| pcnt\_ | percentage |
| pet | potential evapotranspiration |
| \_rcoef\_ | response coefficient |
|  |  |
| pi\_ | plant input |
| precip | precipitation |
| prodn | production |
| prop\_ | proportion |
| rat\_ | ratio |
| \_relse | release |
| scle\_factr | Scaling factor |
| \_sply | supply |
| strss | stress |
| \_ss | steady state |
| t\_ | top soil |
| tair | temperature |
| \_typ, \_atyp | typical, atypical |
| uptk | uptake |
| volat | volatilisation |
| wc\_ | water content |
| iws\_ | water stress index |
| \_yld\_ | yield |
|  |  |

Additional variable naming conventions

|  |  |
| --- | --- |
| hydrol\_eff | Hydrologically effective |
| \_fwd | Forward run |
| \_ss | Steady state |
| pettmp | precipitation, PET, temperature |

# test only

# ========

from numpy import arange

for yld in arange(0.3, 0.9, 0.15):

prop\_n\_opt2 = prop\_n\_optimal\_from\_yield(yld, crop\_vars[crop\_curr])

**Soil nitrogen**

Inputs of nitrate

* Atmospheric deposition
* Fertiliser inputs
* Nitrification

Losses of nitrate

* Immobilisation
* Leaching
* Denitrification
* Crop uptake

Inputs of ammonium

* Atmospheric deposition
* Fertiliser inputs
* Mineralisation

Losses of ammonium

* Immobilisation
* Nitrification
* Volatilisation

**Notes on nomenclature**

Plant production: harvestable yield

Harvest index: is one of the parameters in the model and is C in harvested product over the total C in plant

**Notes on carbon pools**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DPM | pool\_c\_dpm | pi\_to\_dpm | cow\_to\_dpm | c\_loss\_dpm |
| RPM | pool\_c\_rpm | pi\_to\_rpm |  | c\_loss\_rpm |
| HUM | pool\_c\_hum | c\_input\_hum | cow\_to\_hum | c\_loss\_hum |
| BIO | pool\_c\_bio | c\_input\_bio |  | c\_loss\_bio |
| IOM | pool\_c\_iom |  | cow\_to\_iom |  |

**Notes on outputted Excel workbooks**

|  |  |
| --- | --- |
| ora\_classes\_excel\_write.py | Classes defining each workbook |
| ora\_excel\_write.py | Creates sheets using classes in ora\_classes\_excel\_write.py |
| ora\_excel\_write\_cn\_water.py | Writes one workbook for each biophysical group, no class dependencies |