swmmout Documentation

Release 0.1.0

David Townshend

Sep 27, 2017

Contents

swmmout is a simple Python module which reads a SWMM *out* file. It should work on any version of Python >= 2.7, and can be downloaded from PyPI. The source code and issue tracker are on Bitbucket.

CHAPTER 1

Introduction

The main function in this module is `open`, which opens the file specified in its argument and returns a `OutFile` object which can be used to query the file. It is important to note that the file is kept open while the `OutFile` object is in use, and destroying the object will result in the file being closed. It can also be closed forcibly using `OutFile.close`, but this will make all subsequent queries fail.

The main function of the `OutFile` class is `get_values`. This takes five arguments as listed below:

Argument	Optional	Description
group	Required	One of 'subcatchments', 'nodes', 'links', 'system'. This parameter specified which type of objects are to be queried.
names	Optional	An optional list of names to report. If this is omitted, all names found in the specified group will be used. Any unrecognised names are ignored.
variables	Optional	An optional list of variables to report. If this is omitted, all variables found in the specified group will be used. Any unrecognised variables are ignored.
start	Optional	The start time of reporting. This should be a `datetime` object. If it is missing or earlier than the start time of the simulation, that time is used instead.
timesteps	Optional	The number of timesteps to report. If missing, then entire simulation period is returned.

The variable names recognised are as listed below for each group. In addition, any pollutants specified in the model are recognised by prefixing them with 'conc'. For example, if there is a pollutant called 'TSS', an acceptable variables name (for any group) will be 'conc TSS'.

- **subcatchments:**
  - 'rainfall'
  - 'snow depth'
  - 'losses'
  - 'runoff'
  - 'groundwater flow'
  - 'groundwater elevation'
- **nodes:**
  - 'depth'
  - 'head'

  - 'storage'
  - 'lateral inflow'
  - 'total inflow'
  - 'flooding'

links:

  - 'flow'
  - 'depth'
  - 'velocity'
  - 'Froude'
  - 'capacity'

system:

  - 'temperature'
  - 'rainfall'
  - 'snow depth'
  - 'losses'
  - 'runoff'
  - 'dry weather inflow'
  - 'groundwater inflow'
  - 'RDII inflow'
  - 'direct inflow'
  - 'total inflow'
  - 'flooding'
  - 'outflow'
  - 'storage'
  - 'evaporation'

The return value is a list of tuples in the format `[(dateA, nameA, variableA1, variableA2, ...), (dateB, nameB, ...), ...]` This could be seen as a table with columns `Date, Name, Variable1, Variable2, etc`

CHAPTER 2

Example

The following example script shows how to extract selected node information into a csv file:

```python
import csv
import swmmout

# Open the file
outfile = swmmout.open('swmmrun.out')

# Extract some information
query_nodes = ['J1', 'J2', 'J3']
query_vars = ['depth', 'total inflow']
data = outfile.get_values('nodes', query_nodes, query_vars)

# Create a csv file
with csvfile = open('output.csv', 'w'):
    csvwriter = csv.writer(csvfile)
    csvwriter.writerows(data)
```

CHAPTER 3

API

swmmout.**open**(*filename*)

Open a SWMM .out file and return an *OutFile* instance.

class swmmout.**OutFile**(*fh*)

Provide and interface to a SWMM .out file.

The *OutFile* class has methods and properties to allow querying data in a binary SWMM .out file. A typical way of using this by specifying an open file in the constructor, and using *get_values* to query data. For example:

```
>>> import io
>>> swmmfile = io.BytesIO(_test_data)
>>> outfile = OutFile(swmmfile)
>>> outfile.names['nodes']
('J13', 'J16', 'J12', 'J11', 'J15', 'J14', 'J10', 'J9', 'J7', 'J8', 'J4', 'J3',
↪ 'J2', 'J1', 'C8H036', 'Outfall', 'Saulspoort')
>>> outfile.variables['nodes'][0]
'depth'
>>> outfile.start
datetime.datetime(2005, 1, 29, 0, 0)
>>> outfile.get_values('nodes', ['J13', 'J15'], ['depth'], timesteps=3)
[(datetime.datetime(2005, 1, 29, 3, 0), ('J13', 0.0), ('J15', 0.0)),
 (datetime.datetime(2005, 1, 29, 6, 0), ('J13', 0.0), ('J15', 0.0)),
 (datetime.datetime(2005, 1, 29, 9, 0), ('J13', 0.0), ('J15', 0.0))]
```

get_values(*group*, *names=None*, *variables=None*, *start=None*, *timesteps=None*)

Return a list filtered list of values. Output is as a list of tuples in the following format: (datetime, name, variable1, variable2, ...) The list is filtered so that only dates between start and end (inclusive) are in the output.

group is one of 'subcatchments', 'nodes', 'links' or 'system'

close()

Close the file.

Python Module Index

Index