**Pytest with CWATM**

# Purpose

Testing different settings files for different resolutions, different basins and different input and output options in order to make sure that CWatM is properly running after each change.

Folder: pytesting

Program: test\_cwatm3.py

Requirement: libraries pytest and pytest-html

Optional: pip install pytest-codecov

Execute:

pytest test\_cwatm3.py --html=C:/work/CWATM/report1.html  
--settingsfile=test\_py\_cwatm2.txt --cwatm=C:/work/CWATM/run\_cwatm.py

Where:

--html=C:/work/CWATM/report1.html: results of test are written into this HTML file

--settingsfile=test\_py\_cwatm2.txt: settings file to tell which tests are executed

--cwatm=C:/work/CWATM/run\_cwatm.py: executable of CWatM

Or in **PyCharm**

# Files and folders:

output: Folder for CWatM testing results e.g. output/rhine

init: Folder for CWatM warm-start files

settings: Folder with settings templates and created settings files e.g.

Settings/30min/global\_30min/settings\_global\_30min.ini

metaNetcdf.xml: xml file with metadata for NetCDF files

test\_cwatm3.py, conftest.py, pytest.ini: Program files

test\_py\_cwatm1.txt,.. Settings files

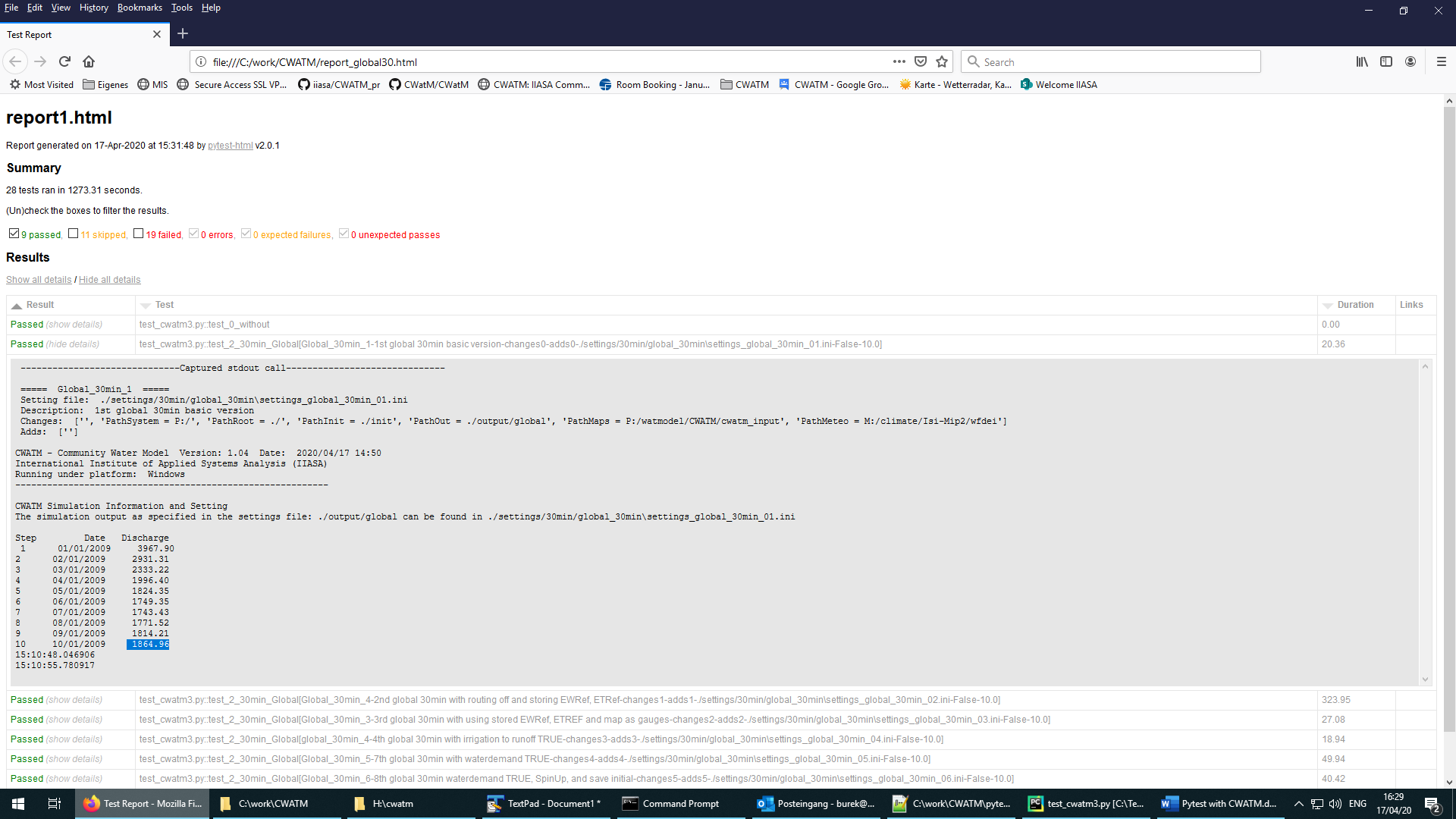
# Testing CWatM

CWatM can be run with different resolutions, different basins, and different options. Each testing e.g. for the Rhine basin 30 arcmin is using a settings template and varies different options during the test. Examples for tests are given below, but other tests can be included. Tests are executed using the template settings file as a start and then tests are repeated using different options.

# Results

Results of the testing are written into the report.html (--html=C:/work/CWATM/report1.html)

This report shows an overview of passed, failed and skipped test and the details of each test.



**Figure 1: Screenshot of a test report page**

# Examples for tests:

1 test mask

1.1 mask as .tif

1.2 mask as box

1.3 mask as outlet lon/lat of a basin

2 test gauges

2.1 as list

2.2 as map

3 Options

3.1 includeIrrigation = True

3.2 preferentialFlow = True

3.3 CapillarRise = True

3.4 includeRunoffConcentration = True

3.5 includeWaterBodies = True => 3.1 - 3.5 together

3.6 includeWaterDemand = True

3.7 calc\_evaporation = True - store OUT\_MAP\_Daily = ETRef, EWRef before!

3.8 includeRouting = False (some more calc\_environflow = True, inflow = True, waterquality = True

4 Timing

4.1 more than 1 year

4.2 with SpinUp

4.3 save initital

4.4 load initial

5 outputs

5.1 with tss output for daily, monthly, yearly

5.2. with map output for daily, monthly, yearly

5.3 with some 'exotic output

# Settingsfile for pytest

This part explains different parts of the settingsfile for pytest e.g. test\_py\_cwatm2.txt

Mind the **;** for separating the changes of the settingsfile (each separated part with a ; is a change in the settingsfile)

# True is the tests for this section should be done

runtest: Rhine\_30min True **-> if this is set to True, test will be executed, otherwise test is skip**

runtest: global\_30min False

runtest: Rhine\_5min False

runtest: global\_5min False

**(at the moment only 4 tests, with several subtest, to be expanded!)**

# True is test should compare last discharge with given number, False is test should look for -no error-

test\_value: False

**-> if this is set to False, a check is done if CWatM runs the settingsfile without error**

**-> if this is set to True, a check is done by comparing the last discharge value with a value in the pytest  
 settingsfile**

# 30 min versions

# Rhine

base\_setting: ./settings/30min/rhine\_30min/settings\_rhine\_30min.ini

* **this is the path to the template CWatM settingsfile. This template file will be modified and executed**

name: Rhine\_30min

* **Name of the test**

path\_system: P:/

path\_root: ./

path\_init: ./init

path\_out: ./output/rhine

path\_maps: P:/watmodel/CWATM/cwatm\_input

path\_meteo: M:/climate/Isi-Mip2/wfdei

* **Main paths of the CWatM settingsfile can be changed here**
* **For the next test only these paths have to be included which have changed since the last test**

# 1st Rhine 30min **-> first sub test**

header: Rhine\_30min\_1 -> name of the test

description: 1st Rhine 30min basic version

set\_save: settings\_rhine\_30min\_01.ini

**-> name of the changed settings file, this file is added in the template settingsfile folder**

changes: **-> changes of the CWatM settingsfile, here: no changes**

adds: **-> add to the CWatM settingsfile, here: no adds**

last\_value: 4.22  **-> last value of discharge to check against if test\_value=True**

….

# 7th setup Rhine **-> sub test no. 7**

header: Rhine\_30min\_7

description: 7th Rhine 30min with waterdemand TRUE

set\_save: settings\_rhine\_30min\_07.ini

changes: calc\_evaporation = False; includeIrrigation = True; preferentialFlow = True; CapillarRise = True;  
 includeRunoffConcentration = True; includeWaterBodies = True; includeWaterDemand = True; StepEnd = 40

adds: OUT\_TSS\_Daily = discharge

last\_value: 443.03

….

# -----------------------------

# Global **-> next test**

base\_setting: ./settings/30min/global\_30min/settings\_global\_30min.ini

name: global\_30min

path\_out: ./output/global

# 1st global 30min

header: Global\_30min\_1

description: 1st global 30min basic version

set\_save: settings\_global\_30min\_01.ini

changes:

adds:

last\_value: 1864.96

# Updating the settingsfile for tests

1. Either you create a new settings file than you need this header

# True is the tests for this section should be done

runtest: Rhine\_30min True

runtest: global\_30min True

# True is the tests for this section should be done

runtest: XXXX\_XXmin True

# True is test should compare last discharge with given number, False is test should look for -no error- execution of run

test\_value: False

# ---------------------

# xx arcmin versions

# Basin xxx

base\_setting: ./settings/XXmin/basin\_XXmin/settings\_XXXX\_XXmin.ini

name: XXXX\_XXmin

path\_system: P:/

path\_root: ./

path\_init: ./init

path\_out: ./output/XXXX

path\_maps: P:/watmodel/CWATM/XXXXXXXXX

path\_meteo: XXXXX

# 1st XXXX XXmin

header: XXXX\_XXmin\_1

description: 1st XXXX XXmin basic version

set\_save: settings\_XXXX\_XXmin\_01.ini

changes:

adds:

last\_value: XXXX

# 2nd setup XXXXX

header: XXXX\_XXmin\_2

description: 2nd XXXX XXmin with box as mask and basic outputs

set\_save: settings\_XXXXX\_XXmin\_02.ini

changes: MaskMap = 14 12 0.5 5.0 52.0

adds: OUT\_TSS\_Daily = discharge; OUT\_MAP\_Daily = discharge

last\_value: XXXX

….

1. Or you add the tests to an existing ones:

* Add runtest: XXXX\_XXmin True below the existing runtest
* Add tests at the bottom

# Settingsfile for CWatM

The template for settingsfile for CWatM can be as individual as needed for specific region, resolutions, options and outputs.

Only the file path sections should contain these keys (key values are changed in the settings file for pytest)

#-------------------------------------------------------

[FILE\_PATHS]

#-------------------------------------------------------

PathSystem = P:/

PathRoot = $(PathSystem)

PathOut = $(PathRoot)/output/

PathInit = $(PathRoot)/init/

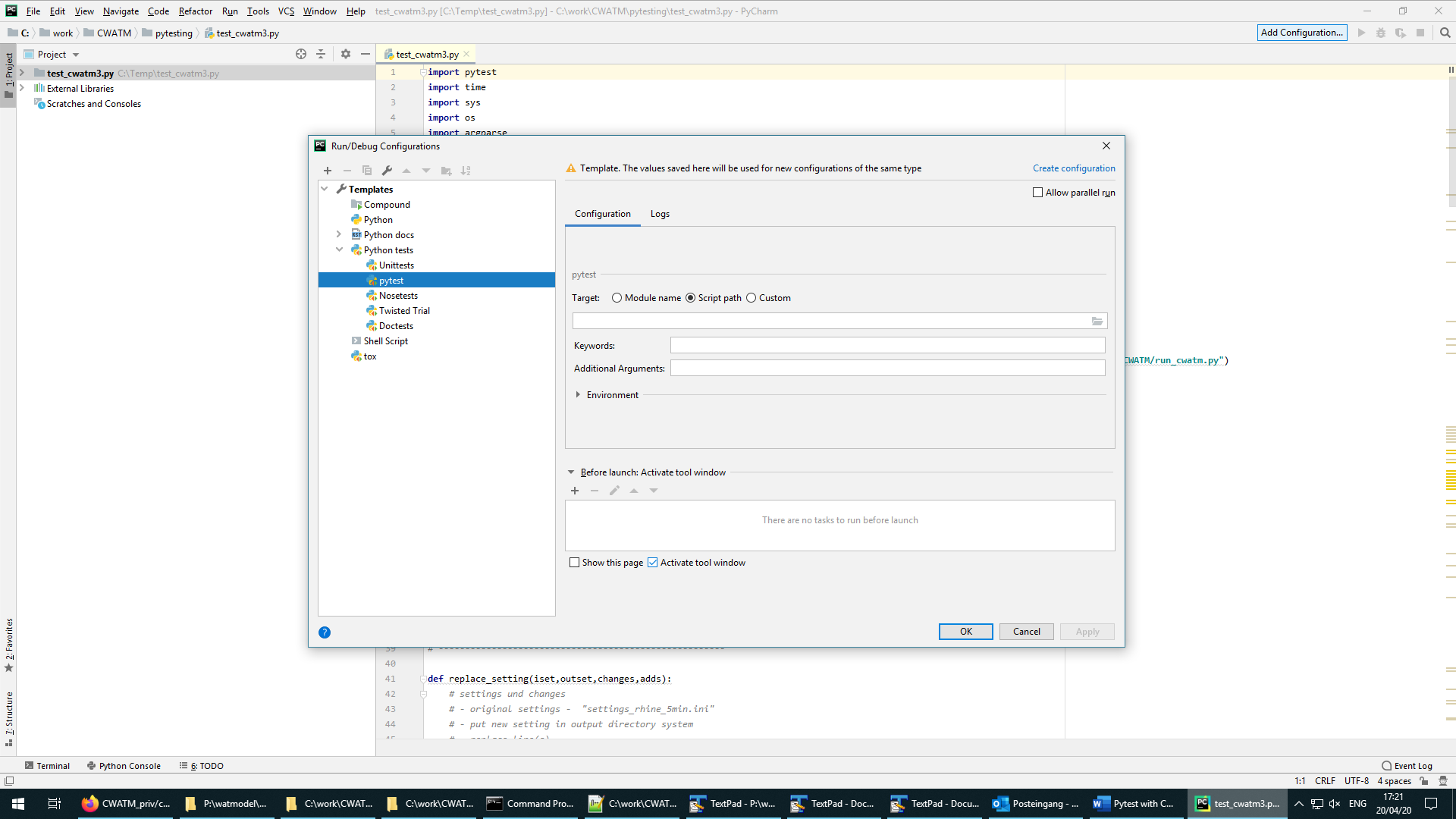
PathMaps = $(PathSystem)/watmodel/CWATM/cwatm\_input

PathWaterdemand = $(PathMaps)/landsurface/waterDemand

PathMeteo = $(PathSystem)

# Executing in PyCharm

1. Load test\_cwatm3.py in Pycharm
2. Open Add configuration (right corner)
3. Look for templates Python test / pytest



1. Insert in Script path:  
   e.g.: C:/work/CWATM/pytesting/test\_cwatm3.py
2. Insert in Additional Arguments:  
   e.g.: --html=report1.html --settingsfile=test\_py\_cwatm2.txt --cwatm=C:/work/CWATM/run\_cwatm.py
3. Right corner of the window: click on set configuration set
4. Run pytest in test\_cwatm3.py (right corner)

