

Setting up PyWPS in a Windows environment with Anaconda

The previous setup of pywps is outdated as it was based on old versions of the WPS 1.0.0 implementation. WPS2.0.0 was launched early 2018. But there are still a lot of applications to be found based on WPS1.0.0 implemented by PyWPS. Here a very short manual with thanks to Maarten Pronk (Deltares) for an easy installation, the former one causes a lot of stress.

Recepies for WPS2 implemented by PyWPS4 on Linux using Python 3.6 will be published later this year. In the meantime it is advised to port your scripts to Python 3.

Step 1 Install Anaconda

Start with installing [Anaconda](#) for Python2.7.

The following packages are required and can be installed via Conda and pip.

```
conda install lxml=3.5.* gdal=1.10.* -c conda-forge
pip install pywps==3.2.6 --no-deps
pip install python-magic
```

Note! This has to be done using a command prompt opened as administrator.

Following steps can be done in any order.

Step 2 Configure PyWPS

1. Create a folder called c:\pywps
2. Create a folder called c:\pywps\processes
3. Download <https://github.com/geopython/pywps/archive/pywps-3.2.6.zip>
4. Extract the file default.cfg from the zip (in pywps) as pywps.cfg in the c:\pywps folder
5. [optional] Extract the files from the pywps/processes folder to c:\pywps\processes to have dummy wps processes for testing
6. [optional] Personalize pywps.cfg with your information
7. Set correct path variables in pywps.cfg
 - a. Set the correct path for tempPath(c:/pywps/temp) under [server] section.
 - b. Set outputPath(c:/[webserver]/htdocs/wps/wpsoutput) under [server] section.
 - c. Set processesPath to c:/pywps/processes
 - d. Set the logFile to somewhere (c:/pywps/wps.log)

The path slashes should conform to python conventions, which happen to be the same as the linux slash convention, and not the windows convention. Note that python paths should never end with a slash!

Step 3 Configure Apache

Navigate to the Apache folder.

If the Apache web server is not installed in your machine yet, download the newest stable release of Apache (<https://httpd.apache.org/download.cgi>).

After installation, in the Apache folder c:\Program Files (x86)\Apache Software Foundation\Apache2.4 you find a sub folder called cgi-bin.

Create an empty file called pywps.cgi in this folder and insert the following code in it, after checking the python path (and correcting it: for example if you have Python2.7 installed, the first line should be #!c:\python27\python.exe): Note for anaconda users this should be #!c:/anaconda/python.exe and the MPLCONFIGDIR should be C:/Anaconda/Lib/site-packages/matplotlib/mpl-data

pywps.cgi

```
#!C:\program files\anaconda2\python.exe
import sys
sys.path.insert(0, r"C:/pywps")
import os
os.environ['PYWPS_CFG']='C:/pywps/pywps.cfg'
os.environ['MPLCONFIGDIR']='C:/Python27/Lib/site-packages/matplotlib/mpl-data'
# Note that WPS runs as apache user and not as you, so many more user variables might be needed.
# Restart Apache after adding one.
import wps
```

Step 4 First test of WPS

Test your WPS by starting Apache server and navigate to <http://localhost/cgi-bin/pywps.cgi> The result should be something similar to this, meaning that the pywps process runs correctly inside the apache web server.

<http://localhost/cgi-bin/pywps.cgi>

```
<ExceptionReport xmlns="http://www.opengis.net/ows/1.1" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.0.0" xsi:schemaLocation="http://www.opengis.net/ows/1.1 http://schemas.opengis.net/ows/1.1.0/owsExceptionReport.xsd">
  <Exception exceptionCode="NoApplicableCode">
    <ExceptionText>'No query string found.'</ExceptionText>
  </Exception>
</ExceptionReport>
```

Step 5 Testing WPS behavior

Now pywps runs, test for correct WPS behavior in browser using the syntax in [WPS Primer](#). In `c:\pywps\tests\processes` some example processes are included, copy its entire contents to the `c:\pywps\pywps_processes` folder (that you configured as `processesPath=c:/pywps/pywps_processes` in `c:\pywps\pywps_processes\default.cfg`) and find out whether you can get them to run. First test the GetCapabilities request:

<http://localhost/cgi-bin/pywps.cgi?service=wps&request=GetCapabilities&version=1.0.0>

```
<wps:Capabilities service="WPS" version="1.0.0" xml:lang="en-CA" xsi:schemaLocation="http://schemas.opengis.net/wps/1.0.0 http://schemas.opengis.net/wps/1.0.0/wpsGetCapabilities_response.xsd" updateSequence="1">
  ...
</wps:Capabilities>
```

and then executing any of the available processes

e.g. [http://localhost/cgi-bin/pywps.cgi?service=wps&request=Execute&Identifier=dummyprocess&DataInputs=\[input1=42\]&version=1.0.0](http://localhost/cgi-bin/pywps.cgi?service=wps&request=Execute&Identifier=dummyprocess&DataInputs=[input1=42]&version=1.0.0).

Step 6 Add wps processes

Now add your own Python processes as WPS by copying your processes to the `c:\pywps\processes` folder too. Use the WPS syntax ([Setting up a WPS process using PyWPS](#)) for your python functions as in the copied examples. Change 3 things when adding a new WPS process based on these examples:

1. add new file `XYZ.py` to `c:/pywps/processes`
2. change identifier inside `XYZ.py`
3. add `XYZ` to the list in `c:/pywps/processes/_init_.py`