

Setting up pyWPS in a Windows environment

NOTE!! This pages is outdated and based on old implementations of PyWPS. The written documentation also often causes problems during installation. Please check the page [Set up PyWPS in a windows environment with Anaconda](#).

The PyWPS software was originally developed to use under Linux (Debian). However, it is also possible to use the software in a Windows environment for example for testing purposes. To do so, Apache server has to be installed on the system. For Linux installation see [this manual](#). Note this pages was initially written for PythonXY users. If you use anaconda bear in mind that location is no longer c:\python2.7 but c:\anaconda. Also mind the version of anaconda installed (x86 or x64 bit version).

- [Step 0 Python installations](#)
- [Step 1 PyWPS](#)
- [Step 2 Install PyWPS](#)
- [Step 3 Configure PyWPS](#)
- [Step 4 Configure Apache](#)
- [Step 5 Install Python PyWPS package](#)
- [Step 6 Install python-magic](#)
- [Step 7 First test of WPS](#)
- [Step 8 Testing WPS behavior](#)
- [Step 9 Add wps processes](#)
- [Known installation issues](#)

Step 0 Python installations

Make sure Python is pre-installed on your machine, we recommend [PythonXY](#) (version 2.7, not 3.x).

After install, start a NEW command line prompt and go to the root and check if :

- Python version is 2.7, Result should be like:

Show Python version

```
C:\>python --version
Python 2.7.6
```

- Python package magic is installed and of version 0.4.6 else see *Known Installation Issues*

Python-Magic package versionnr

```
C:\>pip show python-magic
---
Name: python-magic
Version: 0.4.6
PyPi Version: 0.4.6
Location: c:\python27\lib\site-packages
home_page: http://github.com/ahupp/python-magic
Summary: File type identification using libmagic
Requires:
Required by(0):
```

- Python package lxml is installed and of version 3.3.5 else download <http://www.microsoft.com/en-us/download/details.aspx?id=44266>, install this c++ compiler and install lxml with this command :

```
pip install lxml
```

Python lxml package

```
C:\>pip show lxml
---
Name: lxml
Version: 3.3.5
PyPI Version: 3.4.0
Location: c:\python27\lib\site-packages
home_page: http://lxml.de/
Summary: Powerful and Pythonic XML processing library combining libxml2/libxslt
with the ElementTree API.
Requires:
Extra Require [source]: cython(-)
Extra Require [cssselect]: cssselect
Extra Require [html5]: html5lib
Extra Require [htmlsoup]: BeautifulSoup4
Required by(5): html5lib[optional], pandas[html], astropy[optional], astropy[doc
], BeautifulSoup4[optional]
```

Step 1 PyWPS

[Download PyWPS](#) (version 3.2.1, not 3.2.2)

Step 2 Install PyWPS

Unpack the contents of PyWPS folder and save the entire folder as c:\pywps. Such that you have c:\pywps\wps.py. Check if it works command line in python by navigating to c:\pywps and execute this command:

Test if pywps works on your system

```
C:\>cd pywps

C:\pywps>python wps.py
PyWPS NoApplicableCode: Locator: None; Value: No query string found. Content-Type: application/xml

<?xml version="1.0" encoding="utf-8"?>
<ExceptionReport version="1.0.0" xmlns="http://www.opengis.net/ows/1.1" xmlns:xsi="http://www.w3.org/2001
/XMLSchema-instance" 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 3 Configure PyWPS

1. Create a folder called c:\pywps\pywps_processes
2. Add the file default.cfg (from c:\pywps\pywps) into it. If you like you can insert your own information in the file.
However, do not forget to :
 - 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/pywps_processes
 - d. Set the logFile to somewhere (c:/pywps/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 4 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\Apache2.2 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:\python27\python.exe
import sys
sys.path.insert(0, r"C:/pywps")
import os
os.environ['PYWPS_CFG']='C:/pywps/pywps_processes/default.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 5 Install Python PyWPS package

Install the PyWPS Python package:

- On the command line in the pywps folder `c:\pywps> type python c:\pywps\setup.py install`
- Or open the Windows command line tool (start > run), drag and drop `setup.py` from your `c:\pywps` folder, type `install` after the location of the setup file and click ok. The PyWPS Python package will now be installed.)

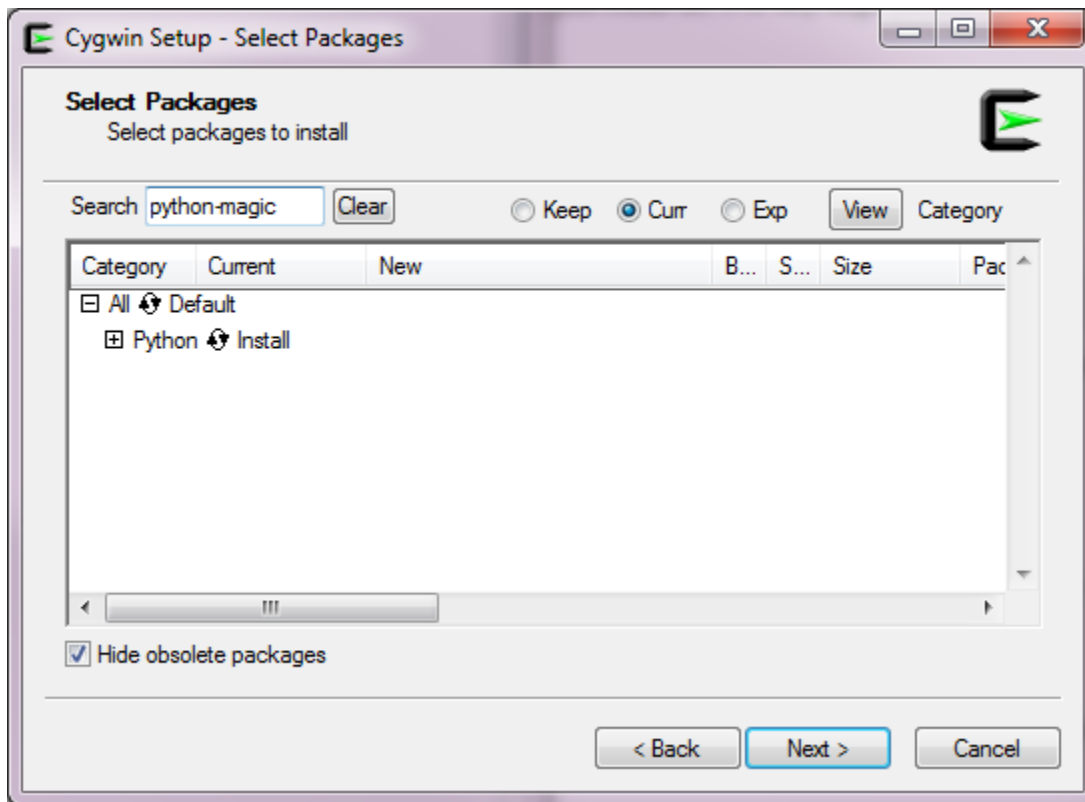
Check if it is installed correctly into python like this :

Show installed python PyWPS version

```
C:\>pip show pywps
---
Name: pywps
Version: 3.2.1
PyPi Version: trunk
Location: c:\python27\lib\site-packages\pywps-3.2.1-py2.7.egg
home_page: http://pywps.wald.intevation.org
Summary: PyWPS: Implementation of OGC's Web Processing Service in Python
Requires:
Required by(0):
```

Step 6 Install python-magic

Check if `cygmagic.dll` or `cygmagic-1.dll` is on your system. If not, install it via installing cygwin (<http://cygwin.com/install.html>). Install the appropriate version (i.e. 32bit or 64bit). You only need to install the python-magic part of cygwin. If you install all, it is very large (several Gb). You can force installation of python-magic by typing this in the search bar, see window below. **NOTE** if python-magic does not yield results. Try **python2-magic** (i.e. Python version for this version of PyWPS).



Set the path to the bin directory of cygwin and copy `cygmagic-1.dll` to `cygmagic.dll`.

NB. If you already have MinGW installed (x86 or x64), you can first try to install `python-magic` without having to install cygwin.

Step 7 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 8 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://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 9 Add wps processes

Now add your own Python processes as WPS by copying your processes to the `c:\pywps\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/pywps_processes`
2. change identifier inside `XYZ.py`
3. add `"XYZ.py"` to `c:/pywps/pywps_processes/_init_.py`

Known installation issues

- If the GetCapabilities request gives you this error

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

Internal Server Error

in your html browser, please check your apache error.log in `c:\Program Files (x86)\Apache Software Foundation\Apache2.2\logs`.

- In some cases it might contain this error message `ImportError: No module named magic\r`. In this case please manually install the python-magic package from <https://github.com/ahupp/python-magic>. For your convenience we included a windows distributable [win32 Python2.7](#) version until the official binary is available at github. Restart apache afterwards (type services in your windows start menu or click small apache icon in icon tray). If you still get errors (failed to find libmagic in the aforementioned Apache error.log) install cygwin <http://cygwin.com/install.html> which contains the magic dll. Set the path where the `cygmagic-1.dll` resides `c:\Program Files (x86)\Cygwin\bin` in the windows \$PATH environment variable so magic.py can find it (bear in mind that this should be the System Variables Path setting). Now several options are available to solve this issue, these are:
 - If you don't want to have cygwin on your machine just copy the following dynamic link library (dll) files from the cygwin installation to the python root directory (where python.exe resides) and uninstall / remove cygwin:
 - `cygmagic-1.dll` (rename this to `cygmagic.dll`!!!!!!)
 - `cyggcc_s-1.dll`
 - `cygwin1.dll`
 - `cygz.dll`
 - Pay attention to the CPU Type installation, if you have installed a 64 bits python use 64 bits libs!!
 - Final trick in case of , if all off the above does not help and your made sure that you have all correct libraries installed for the correct CPU type, you could alter the file called magic.py. Dependent on the type of python installer (for now Anaconda and PythonXY are known to the different authors of this page) this magic.py is available in the `<pythonpath>\lib\site-packages` directory. Search for the line called `'platform_to_lib ='` it should mention between `[] 'magic1.dll'`. Complete that list (python list) with `'cygmagic.dll'`.
- In some cases it might contain this error message : `ImportError: No module named Template\r\r` ; You probably have installed PyWPS version 3.2.2. Please remove this version and download and install version 3.2.1.
- The following error `AttributeError: 'exceptions.ImportError' object has no attribute 'replace'\r` is caused by the fact that step 7 is not fully followed. So put some processes in your processes folder and try the GetCapabilities again.
- If you run this in a Virtual windows machine (e.g. via VMWare), regular disk mappings might not work in WPS because WPS runs as another users (apache), please try references like `//vmware-host/Shared%20Folders/D/` instead of simply `D`.
- Errors like these for import statements in a WPS process, e.g. `import matplotlib` causes

```
PyWPS [2014-03-19 14:58:23,112] WARNING: Could not import processes from 'pywps_processes': ImportError ('No module named pwd',)
```

have to do with the fact that WPS runs under apache user, so add any required user environmental variables to `c:\Program Files (x86)\Apache Software Foundation\Apache2.2\cgi-bin\pywps.cgi`:

```
pywps.cgi
```

```
os.environ['MPLCONFIGDIR']='C:/Python27/Lib/site-packages/matplotlib/mpl-data'
```

Notes:

- PyWPS can also be used with the [Q-GIS WPS plugin](#) by Dr. Horst Duester.
- Official documentation on PyWPS (v3.2.0) can be found [here](#)
- To install basemap you can follow the instructions [here](#)

Acknowledgements The first version of this documentation was made by MSc student Joost Boerboom for his [thesis](#), financed by Deltares under the [Next Generation Hydro Software](#) program executed for Rijkswaterstaat. Subsequently, this documentation was improved by Deltares in projects [EMODnet chemistry](#), [ECOSTRESS](#) and [FAST](#).