

---

# **CliMAF WPS Demo Documentation**

***Release 1.1.1***

**Carsten Ehbrecht**

**Aug 02, 2019**



---

## Contents:

---

<b>1 Credits</b>	<b>3</b>
<b>2 Indices and tables</b>	<b>9</b>
<b>Index</b>	<b>11</b>



A Web Processing Service for CliMAF.

- Free software: Apache Software License 2.0
- Documentation: <https://climaf-wps-demo.readthedocs.io>.



This package was created with [Cookiecutter](#) and the [bird-house/cookiecutter-birdhouse](#) project template.

## 1.1 Installation

- *Install from Conda*
- *Install from GitHub*
- *Start CliMAF WPS Demo PyWPS service*
- *Run CliMAF WPS Demo as Docker container*
- *Use Ansible to deploy CliMAF WPS Demo on your System*

### 1.1.1 Install from Conda

**Warning:** TODO: Prepare Conda package.

### 1.1.2 Install from GitHub

Check out code from the CliMAF WPS Demo GitHub repo and start the installation:

```
$ git clone https://github.com/cp4cds/climaf-wps-demo.git
$ cd climafwps
$ conda env create -f environment.yml
$ source activate climafwps
$ python setup.py develop
```

### ... or do it the lazy way

The previous installation instructions assume you have Anaconda installed. We provide also a Makefile to run this installation without additional steps:

```
$ git clone https://github.com/cp4cds/climaf-wps-demo.git
$ cd climafwps
$ make clean      # cleans up a previous Conda environment
$ make install    # installs Conda if necessary and runs the above installation steps
```

### 1.1.3 Start CliMAF WPS Demo PyWPS service

After successful installation you can start the service using the `climafwps` command-line.

```
$ climafwps --help # show help
$ climafwps start  # start service with default configuration

OR

$ climafwps start --daemon # start service as daemon
loading configuration
forked process id: 42
```

The deployed WPS service is by default available on:

<http://localhost:5000/wps?service=WPS&version=1.0.0&request=GetCapabilities>.

---

**Note:** Remember the process ID (PID) so you can stop the service with `kill PID`.

---

You can find which process uses a given port using the following command (here for port 5000):

```
$ netstat -nlp | grep :5000
```

Check the log files for errors:

```
$ tail -f pywps.log
```

### ... or do it the lazy way

You can also use the Makefile to start and stop the service:

```
$ make start
$ make status
$ tail -f pywps.log
$ make stop
```

### 1.1.4 Run CliMAF WPS Demo as Docker container

You can also run CliMAF WPS Demo as a Docker container.



**Warning:** TODO: Describe Docker container support.

## 1.1.5 Use Ansible to deploy CliMAF WPS Demo on your System

Use the [Ansible playbook](#) for PyWPS to deploy CliMAF WPS Demo on your system.

## 1.2 Configuration

### 1.2.1 Command-line options

You can overwrite the default [PyWPS](#) configuration by using command-line options. See the CliMAF WPS Demo help which options are available:

```
$ climafwps start --help
--hostname HOSTNAME      hostname in PyWPS configuration.
--port PORT               port in PyWPS configuration.
```

Start service with different hostname and port:

```
$ climafwps start --hostname localhost --port 5001
```

### 1.2.2 Use a custom configuration file

You can overwrite the default [PyWPS](#) configuration by providing your own PyWPS configuration file (just modify the options you want to change). Use one of the existing `sample-*.cfg` files as example and copy them to `etc/custom.cfg`.

For example change the hostname (*localhost*) and the path to the CMIP5 data archive:

```
$ cd climafwps
$ vim etc/custom.cfg
$ cat etc/custom.cfg
[server]
url = http://localhost:5000/wps
outputurl = http://localhost:5000/outputs

[logging]
level = INFO

[data]
archive_root = /data/cmip5/data
```

Start the service with your custom configuration:

```
# start the service with this configuration
$ climafwps start -c etc/custom.cfg
```

## 1.3 Developer Guide

- *Building the docs*
- *Running tests*
- *Run tests the lazy way*
- *Prepare a release*
- *Bump a new version*

**Warning:** To create new processes look at examples in [Emu](#).

### 1.3.1 Building the docs

First install dependencies for the documentation:

```
$ make bootstrap_dev
$ make docs
```

### 1.3.2 Running tests

Run tests using `pytest`.

First activate the `climafwps` Conda environment and install `pytest`.

```
$ source activate climafwps
$ conda install pytest flake8 # if not already installed
```

Run quick tests (skip slow and online):

```
$ pytest -m 'not slow and not online''
```

Run all tests:

```
$ pytest
```

Check pep8:

```
$ flake8
```

### 1.3.3 Run tests the lazy way

Do the same as above using the Makefile.

```
$ make test
$ make testall
$ make pep8
```

### 1.3.4 Prepare a release

Update the Conda specification file to build identical [environments](#) on a specific OS.

**Note:** You should run this on your target OS, in our case Linux.

```
$ make clean
$ make install
$ make spec
```

### 1.3.5 Bump a new version

Make a new version of CliMAF WPS Demo in the following steps:

- Make sure everything is commit to GitHub.
- Update `CHANGES.rst` with the next version.
- Dry Run: `bumpversion --dry-run --verbose --new-version 0.8.1 patch`
- Do it: `bumpversion --new-version 0.8.1 patch`
- ... or: `bumpversion --new-version 0.9.0 minor`
- Push it: `git push`
- Push tag: `git push --tags`

See the [bumpversion](#) documentation for details.

## 1.4 Processes

- *Sleep*
- *CMIP5 Global Mean Time Series*

### 1.4.1 Sleep

```
class climafwps.processes.wps_sleep.Sleep
    sleep Sleep Process (v1.0)
```

Testing a long running process, in the sleep. This process will sleep for a given delay or 10 seconds if not a valid value.

**Parameters** `delay` (`{ 'None' }`) – Delay between every update

**Returns** `sleep_output` – Sleep Output

**Return type** string

#### References

- [PyWPS Demo](#)

## 1.4.2 CMIP5 Global Mean Time Series

**class** climafwps.processes.wps\_tsplot.**TimeSeriesPlot**  
**tsplot** CMIP5 Global Mean Time Series (v1.1.1)

Uses the CliMAF tool to calculate a time series of global mean values for a variable, model, experiment and ensemble member from the CMIP5 archive. The time series is plotted as a line graph showing change in the global mean value against time.

### Parameters

- **model** ({'ACCESS1-0', 'ACCESS1-3', 'bcc-csm1-1', 'bcc-csm1-1-m', 'BNU-ESM', 'CanCM4', 'CanESM2', 'CCSM4', 'CESM1-BGC', 'CESM1-CAM5', ..}) – Climate model ID
- **experiment** ({'rcp45', 'rcp60', 'rcp8', 'historical'}) – Experiment name
- **variable** ({'cl', 'clt', 'evspsbl', 'hur', 'huss', 'prc', 'ps', 'rldscs', 'rlutcs', 'rsdt', ..}) – Variable ID
- **start\_year** ({'None'}) – 4-digit start year
- **end\_year** ({'None'}) – 4-digit end year

**Returns** **output** – Generated timeseries plot.

**Return type** *image/png*

### References

- [CliMAF](#)
- [Documentation](#)
- [Media](#)

## 1.5 Changes

### 1.5.1 1.1.1 (2019-08-02)

- Updated from latest cookiecutter template (#6).
- Fixed conda env (#5)

### 1.5.2 1.1.0 (2018-07-11)

- First release.

## CHAPTER 2

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`



## S

Sleep (*class in climafwps.processes.wps\_sleep*), [7](#)

## T

TimeSeriesPlot (*class in climafwps.processes.wps\_tsplot*), [8](#)