c3s magic wps Documentation

Release 1.0.0

Berend Weel

Jul 19, 2019

Contents:

Web Processing Service for Climate Data Analysis in the MAGIC project. The software in this WPS powers the processes behind the C3S-MAGIC portal. The processes are run on CP4CDS infrastructure.

- Free software: Apache Software License 2.0
- Documentation: https://c3s-magic-wps.readthedocs.io.

CHAPTER 1

Links

- Climate Data Store
- C3S MAGIC Portal
- CP4CDS Quality Control

CHAPTER 2

Credits

This package was created with Cookiecutter and the bird-house/cookiecutter-birdhouse project template.

2.1 Installation

2.1.1 Install from PyPI

The MAGIC WPS is available as a package in *PyPI <https://pypi.org/>* (c3s-magic-wps). A working installation of ESMValTool is required, currently version 2.0a2 is supported.

2.1.2 Install from GitHub

Note: These installation instructions assume you have Anaconda *installed.*

Note: these installtion instructions assume you have Julia *installed*

Check out code from the c3s magic wps GitHub repo and create a conda environment:

```
$ git clone https://github.com/c3s-magic/c3s-magic-wps.git
$ cd c3s-magic-wps
$ conda env create -f environment.yml
$ source activate c3s_magic_wps
```

Note: the environment will pull pywps and esmvaltool from github via pip

Next, to complete the installation of ESMValTool a R and Julia script are required. These are available in the package. Please adjust to match the installation location of conda on your system

```
$ Rscript /opt/conda/envs/c3s-magic-wps/lib/python3.6/site-packages/esmvaltool/
↪install/R/setup.R
$ julia /opt/conda/envs/c3s-magic-wps/lib/python3.6/site-packages/esmvaltool/install/
↪Julia/setup.jl
```

Finally install the WPS:

```
$ cd ../c3s-magic-wps
$ python setup.py develop
```

2.1.3 Configure c3s magic wps PyWPS service

The wps can be configured using the pywps configuration files. See the etc folder for examples. Create a file called `.custom.cfg` to customize settings for your installation. A path to the cmip and obs files is needed to run the metrics.

See the *Installation* section for more info

2.1.4 Start c3s magic wps PyWPS service

After successful installation you can start the service using the `c3s_magic_wps` command-line. An additional environment variable is needed with the location of the model data.

```
$ export CMIP_DATA_ROOT=/path/to/cmip/files


$ c3s_magic_wps --help # show help
$ c3s_magic_wps start  # start service with default configuration

OR

$ c3s_magic_wps start --daemon # start service as daemon
loading configuration
forked process id: 42
```

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

The deployed WPS service is by default available on:

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

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
```

2.1.5 Run c3s magic wps as Docker container

Note: These installation instructions assume you have Docker *installed.*

You can also choose to run c3s magic wps from a Docker container.

Download c3s-magic-wps, build the docker container and run it using docker-compose:

```
$ git clone https://github.com/c3s-magic/c3s-magic-wps.git
$ cd c3s-magic-wps
$ docker-compose build
$ docker-compose up
```

By default the WPS service should be available on port 5000:

http://localhost:5000/wps?service=wps&request=GetCapabilities

Run docker exec to watch logs:

```
$ docker ps      # find container name
container_name
$ docker exec container_name tail -f /opt/wps/pywps.log
```

Use docker-compose to stop the containers:

```
$ docker-compose down
```

2.1.6 Use Ansible to deploy c3s magic wps on your System

Use the Ansible playbook for PyWPS to deploy c3s magic wps on your system.

2.2 Configuration

2.2.1 Command-line options

You can overwrite the default PyWPS configuration by using command-line options. See the c3s magic wps help which options are available:

```
$ c3s_magic_wps start --help
--hostname HOSTNAME        hostname in PyWPS configuration.
--port PORT                port in PyWPS configuration.
```

Start service with different hostname and port:

```
$ c3s_magic_wps start --hostname localhost --port 5001
```

2.2.2 Use a custom configuration file

You can overwrite the default PyWPS configuration by providing your own PyWPS configuration file (just modifiy 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 (*demo.org*) and logging level:

```
$ cd c3s_magic_wps
$ vim etc/custom.cfg
$ cat etc/custom.cfg
[server]
url = http://demo.org:5000/wps
outputurl = http://demo.org:5000/outputs

[logging]
level = DEBUG

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

Start the service with your custom configuration:

```
# start the service with this configuration
$ c3s_magic_wps start -c etc/custom.cfg
```

2.3 Processes

2.3.1 Data Availability

A special *Meta* process in the Magic WPS is available to automatically determine the available cmip5 model data by reading the available files from the data folder. As CMIP5 data is structured according to the DRS this can be done automatically. The format used to pass this information is the json output of the linux tree command. If configured correctly, the WPS processes will automatically find the data

```
tree -J -l -d -L 8 /group_workspaces/jasmin2/cp4cds1/data/c3s-cmip5
```

2.3.2 Metrics

This is a list of metrics available as processes in the MAGIC WPS.

The list of Models, Experiments, and Ensembles is created automatically when the server is started, and shown as "None" here.

class c3s_magic_wps.processes.**Blocking**
Bases: pywps.app.Process.Process

blocking Blocking metrics and indices (v1.0.0)

Calculate Blocking metrics that shows the mid-latitude 1D and 2D blocking indices.

Parameters

- **model** (*{'None'}*) – Choose a model
- **experiment** (*{'None'}*) – Choose an experiment
- **ensemble** (*{'None'}*) – Choose an ensemble
- **start_year** (*integer*) – Start year of model data.
- **end_year** (*integer*) – End year of model data.
- **ref_dataset** (*{'ERA-Interim'}*) – Choose a reference dataset like ERA-Interim.
- **season** (*{'DJF', 'MAM', 'JJA', 'SON', 'ALL'}*) – Choose a season like DJF.

Returns

- **tm90_plot** (*image/png*) – Generated TM90 plot of ESMValTool processing.
- **numberevents_plot** (*image/png*) – Generated NumberEvents plot of ESMValTool processing.
- **durationevents_plot** (*image/png*) – Generated DurationEvents plot of ESMValTool processing.
- **longblockevents_plot** (*image/png*) – Generated LongBlockEvents plot of ESMValTool processing.
- **blockevents_plot** (*image/png*) – Generated BlockEvents plot of ESMValTool processing.
- **acn_plot** (*image/png*) – Generated ACN plot of ESMValTool processing.
- **cn_plot** (*image/png*) – Generated CN plot of ESMValTool processing.
- **bi_plot** (*image/png*) – Generated BI plot of ESMValTool processing.
- **mgi_plot** (*image/png*) – Generated MGI plot of ESMValTool processing.
- **z500_plot** (*image/png*) – Generated Z500 plot of ESMValTool processing.
- **extrablock_plot** (*image/png*) – Generated ExtraBlock plot of ESMValTool processing.
- **instblock_plot** (*image/png*) – Generated InstBlock plot of ESMValTool processing.
- **data_full** (*application/x-netcdf*) – Generated output data of ESMValTool processing.
- **data_clim** (*application/x-netcdf*) – Generated output data of ESMValTool processing.
- **archive** (*application/zip*) – The complete output of the ESMValTool processing as an zip archive.
- **success** (*string*) – True if the metric has been successfully calculated. If false please check the log files
- **recipe** (*text/plain*) – ESMValTool recipe used for processing.
- **log** (*text/plain*) – Log File of ESMValTool processing.
- **debug_log** (*text/plain*) – Debug Log File of ESMValTool processing.

References

- ESMValTool
- Documentation
- Estimated Calculation Time

get_outputs (*result*, *subdir*, *response*)

class c3s_magic_wps.processes.**CVDP**
Bases: pywps.app.Process.Process

cvdp NCAR CVDPackage (v1.0.0)

Run the NCAR CVDPackage

Parameters

- **model** (*{'None'}*) – Choose a model
- **experiment** (*{'None'}*) – Choose an experiment
- **ensemble** (*{'None'}*) – Choose an ensemble
- **start_year** (*integer*) – Start year of model data.
- **end_year** (*integer*) – End year of model data.

Returns

- **tas_trend_ann_plot** (*image/png*) – Annual trend in surface air temperature.
- **sst_trend_ann_plot** (*image/png*) – Annual trend in sea surface temperature.
- **psl_trend_ann_plot** (*image/png*) – Annual trend in sea level pressure.
- **pr_trend_ann_plot** (*image/png*) – Annual trend in precipitation.
- **archive** (*application/zip*) – The complete output of the ESMValTool processing as an zip archive.
- **success** (*string*) – True if the metric has been successfully calculated. If false please check the log files
- **recipe** (*text/plain*) – ESMValTool recipe used for processing.
- **log** (*text/plain*) – Log File of ESMValTool processing.
- **debug_log** (*text/plain*) – Debug Log File of ESMValTool processing.

References

- ESMValTool
- Documentation
- Estimated Calculation Time

get_outputs (*result*, *response*)

class c3s_magic_wps.processes.**CapacityFactor**
Bases: pywps.app.Process.Process

capacity_factor Capacity factor of wind power (v1.0.0)

The goal of this diagnostic is to compute the wind capacity factor, taking as input the daily instantaneous surface wind speed, which is then extrapolated to obtain the wind speed at a height of 100 m as described in Lledó (2017).

Parameters

- **model** (*{'None'}*) – Choose a model
- **experiment** (*{'None'}*) – Choose an experiment
- **ensemble** (*{'None'}*) – Choose an ensemble
- **start_year** (*integer*) – Start year of model data.
- **end_year** (*integer*) – End year of model data.
- **start_longitude** (*integer*) – Minimum longitude.
- **end_longitude** (*integer*) – Maximum longitude.
- **start_latitude** (*integer*) – Minimum latitude.
- **end_latitude** (*integer*) – Maximum latitude.
- **season** (*{'DJF', 'MAM', 'JJA', 'SON'}*) – Season

Returns

- **plot** (*image/png*) – Ratio of average estimated power to theoretical maximum power.
- **data** (*application/zip*) – Ratio of average estimated power to theoretical maximum power.
- **archive** (*application/zip*) – The complete output of the ESMValTool processing as an zip archive.
- **success** (*string*) – True if the metric has been successfully calculated. If false please check the log files
- **recipe** (*text/plain*) – ESMValTool recipe used for processing.
- **log** (*text/plain*) – Log File of ESMValTool processing.
- **debug_log** (*text/plain*) – Debug Log File of ESMValTool processing.

References

- ESMValTool
- Documentation
- Estimated Calculation Time

get_outputs(*result*, *response*)

class c3s_magic_wps.processes.**CombinedIndices**

Bases: pywps.app.Process.Process

combined_indices Single and multi-model indices based on area averages (v1.0.0)

Metric showning single and multi model indices based on area averages.

Parameters

- **model** (*{'None'}*) – Choose a model
- **experiment** (*{'None'}*) – Choose an experiment

- **ensemble** (*{'None'}*) – Choose an ensemble
- **start_year** (*integer*) – Start year of model data.
- **end_year** (*integer*) – End year of model data.
- **running_mean** (*integer*) – Length of the window for which the running mean is computed.
- **moninf** (*{'1', '2', '3', '4', '5', '6', '7', '8', '9', '10', . .}*) – First month of the seasonal mean period to be computed, if null the monthly anomalies will be computed.
- **monsup** (*{'1', '2', '3', '4', '5', '6', '7', '8', '9', '10', . .}*) – Last month of the seasonal mean period to be computed, if null the monthly anomalies will be computed.
- **region** (*{'NAO', 'Nino3', 'Nino3.4', 'Nino4', 'SOI'}*) – Region over which to calculate the metric.
- **standardized** (*boolean*) – Boolean indictating if standardization should be computed.

Returns

- **plot** (*image/png*) – Combined Indices plot.
- **data** (*application/x-netcdf*) – Generated combined indices data.
- **archive** (*application/zip*) – The complete output of the ESMValTool processing as an zip archive.
- **success** (*string*) – True if the metric has been successfully calculated. If false please check the log files
- **recipe** (*text/plain*) – ESMValTool recipe used for processing.
- **log** (*text/plain*) – Log File of ESMValTool processing.
- **debug_log** (*text/plain*) – Debug Log File of ESMValTool processing.

References

- ESMValTool
- Documentation
- Estimated Calculation Time

get_outputs(*result*, *response*)

class c3s_magic_wps.processes.**ConsecDryDays**

Bases: pywps.app.Process.Process

consecdrydays Consecutive Dry Days (v1.0.0)

Calculates the longest period of consecutive dry days (days with at least 'prlim' mm/day) in the provided time series, as well as the number of periods of at least 'frlim' consecutive dry days. 'prlim' and 'frlim' are provided by the user.

Parameters

- **model** (*{'None'}*) – Choose a model
- **experiment** (*{'None'}*) – Choose an experiment

- **ensemble** (*{'None'}*) – Choose an ensemble
- **start_year** (*integer*) – Start year of model data.
- **end_year** (*integer*) – End year of model data.
- **frlim** (*{'2.5', '5', '10'}*) – The shortest number of consecutive dry days for entering statistic on frequency of dry periods.
- **plim** (*{'0.5', '1', '2'}*) – Limit for a day to be considered dry [mm/day].

Returns

- **dryfreq_plot** (*image/png*) – Generated dryfreq plot of ESMValTool processing.
- **drymax_plot** (*image/png*) – Generated drymax plot of ESMValTool processing.
- **data_drymax** (*application/x-netcdf*) – Generated output data of ESMValTool processing.
- **data_dryfreq** (*application/x-netcdf*) – Generated output data of ESMValTool processing.
- **archive** (*application/zip*) – The complete output of the ESMValTool processing as an zip archive.
- **success** (*string*) – True if the metric has been successfully calculated. If false please check the log files
- **recipe** (*text/plain*) – ESMValTool recipe used for processing.
- **log** (*text/plain*) – Log File of ESMValTool processing.
- **debug_log** (*text/plain*) – Debug Log File of ESMValTool processing.

References

- ESMValTool
- Documentation
- Media
- Estimated Calculation Time

get_outputs(*result*, *response*)

class c3s_magic_wps.processes.**DiurnalTemperatureIndex**

Bases: pywps.app.Process.Process

diurnal_temperature_index Diurnal Temperature Variation (DTR) Indicator (v1.0.0)

Metric showing the diurnal temperature indicator to estimate energy demand.

Parameters

- **model** (*{'None'}*) – Choose a model
- **experiment** (*{'None'}*) – Choose an experiment
- **ensemble** (*{'None'}*) – Choose an ensemble
- **start_historical** (*integer*) – Start historical of model data.
- **end_historical** (*integer*) – End historical of model data.
- **start_projection** (*integer*) – Start projection of model data.

- **end_projection** (*integer*) – End projection of model data.
- **start_longitude** (*integer*) – Minimum longitude.
- **end_longitude** (*integer*) – Maximum longitude.
- **start_latitude** (*integer*) – Minimum latitude.
- **end_latitude** (*integer*) – Maximum latitude.

Returns

- **plot** (*image/png*) – The diurnal temperature indicator to estimate energy demand.
- **data** (*application/zip*) – The diurnal temperature indicator data.
- **archive** (*application/zip*) – The complete output of the ESMValTool processing as an zip archive.
- **success** (*string*) – True if the metric has been successfully calculated. If false please check the log files
- **recipe** (*text/plain*) – ESMValTool recipe used for processing.
- **log** (*text/plain*) – Log File of ESMValTool processing.
- **debug_log** (*text/plain*) – Debug Log File of ESMValTool processing.

References

- ESMValTool
- Documentation
- Media
- Estimated Calculation Time

get_outputs(*result*, *response*)

class c3s_magic_wps.processes.**DroughtIndicator**

Bases: pywps.app.Process.Process

drought_indicator Drought indicator (v1.0.0)

The drought indicator calculates diagnostics for meteorological drought.

Parameters

- **model** (*{'None'}*) – Choose a model
- **experiment** (*{'None'}*) – Choose an experiment
- **ensemble** (*{'None'}*) – Choose an ensemble
- **start_year** (*integer*) – Start year of model data.
- **end_year** (*integer*) – End year of model data.
- **ref_dataset** (*{'ERA-Interim'}*) – Choose a reference dataset like ERA-Interim.

Returns

- **spi_plot** (*image/png*) – Generated spi histogram plot.
- **spei_plot** (*image/png*) – Generated SPEI Histogram plot.
- **spi_model** (*application/x-netcdf*) – The complete SPI Data for the model.

- **spi_reference** (*application/x-netcdf*) – The complete SPI Data for the reference model.
- **spei_model** (*application/x-netcdf*) – The complete SPEI Data for the model.
- **spei_reference** (*application/x-netcdf*) – The complete SPEI Data for the reference model.
- **archive** (*application/zip*) – The complete output of the ESMValTool processing as an zip archive.
- **success** (*string*) – True if the metric has been successfully calculated. If false please check the log files
- **recipe** (*text/plain*) – ESMValTool recipe used for processing.
- **log** (*text/plain*) – Log File of ESMValTool processing.
- **debug_log** (*text/plain*) – Debug Log File of ESMValTool processing.

References

- ESMValTool
- Documentation
- Estimated Calculation Time

get_outputs (*result*, *response*)

class c3s_magic_wps.processes.**EnsClus**

Bases: pywps.app.Process.Process

ensclus EnsClus - Ensemble Clustering (v1.0.0)

Cluster analysis tool based on the k-means algorithm for ensembles of climate model simulations. EnsClus group ensemble members according to similar characteristics and select the most representative member for each cluster.

Parameters

- **model** (*{'None'}*) – Choose a model
- **experiment** (*{'None'}*) – Choose an experiment
- **ensemble** (*{'None'}*) – Choose an ensemble
- **start_year** (*integer*) – Start year of model data.
- **end_year** (*integer*) – End year of model data.
- **variable** (*{'pr', 'tas'}*) – Select the variable to simulate.
- **season** (*{'DJF', 'DJFM', 'NDJFM', 'JJA'}*) – Choose a season like DJF.
- **area** (*{'EU', 'EAT', 'PNA', 'NH'}*) – Area over which to calculate.
- **extreme** (*{'60th_percentile', '75th_percentile', '90th_percentile', 'mean', 'maximum', 'std', 'trend'}*) – Extreme metric.
- **numclus** (*integer*) – Number of clusters.
- **perc** (*{'70', '80', '90'}*) – Percentage of total Variance

- **numpcs** (*integer*) – Number of PCs to retain. Has priority over Percentage unless set to 0

Returns

- **plot** (*image/eps*) – Generated output plot of ESMValTool processing.
- **ens_extreme** (*application/x-netcdf*) – Generated output data of ESMValTool processing.
- **ens_climatologies** (*application/x-netcdf*) – Generated output data of ESMValTool processing.
- **ens_anomalies** (*application/x-netcdf*) – Generated output data of ESMValTool processing.
- **statistics** (*text/plain*) – Clustering Statistics
- **archive** (*application/zip*) – The complete output of the ESMValTool processing as an zip archive.
- **success** (*string*) – True if the metric has been successfully calculated. If false please check the log files
- **recipe** (*text/plain*) – ESMValTool recipe used for processing.
- **log** (*text/plain*) – Log File of ESMValTool processing.
- **debug_log** (*text/plain*) – Debug Log File of ESMValTool processing.

References

- ESMValTool
- Documentation
- Media
- Model Selection
- Estimated Calculation Time

get_outputs (*result*, *response*)

class c3s_magic_wps.processes.**ExtremeIndex**

Bases: pywps.app.Process.Process

extreme_index Combined Climate Extreme Index (v1.0.0)

Metric showing extreme indices relevant to the insurance industry (heat, cold, wind, flood and drought indices).

Parameters

- **model** (*{'None'}*) – Choose a model
- **experiment** (*{'None'}*) – Choose an experiment
- **ensemble** (*{'None'}*) – Choose an ensemble
- **start_historical** (*integer*) – Start historical of model data.
- **end_historical** (*integer*) – End historical of model data.
- **start_projection** (*integer*) – Start projection of model data.
- **end_projection** (*integer*) – End projection of model data.

- **running_mean** (*integer*) – Length of the window for which the running mean is computed.
- **start_longitude** (*integer*) – Minimum longitude.
- **end_longitude** (*integer*) – Maximum longitude.
- **start_latitude** (*integer*) – Minimum latitude.
- **end_latitude** (*integer*) – Maximum latitude.

Returns

- **t10p_plot** (*image/png*) – Generated t10p plot of ESMValTool processing.
- **t90p_plot** (*image/png*) – Generated t90p plot of ESMValTool processing.
- **wx_plot** (*image/png*) – Generated Wx plot of ESMValTool processing.
- **rx5day_plot** (*image/png*) – Generated rx5day plot of ESMValTool processing.
- **cdd_plot** (*image/png*) – Generated cdd plot of ESMValTool processing.
- **combined_plot** (*image/png*) – Generated combined plot of ESMValTool processing.
- **t10p_data** (*application/x-netcdf*) – Generated output t10p data of ESMValTool processing..
- **t90p_data** (*application/x-netcdf*) – Generated output t90p data of ESMValTool processing..
- **wx_data** (*application/x-netcdf*) – Generated output Wx data of ESMValTool processing..
- **rx5day_data** (*application/x-netcdf*) – Generated output rx5day data of ESMValTool processing..
- **cdd_data** (*application/x-netcdf*) – Generated output cdd data of ESMValTool processing..
- **combined_data** (*application/x-netcdf*) – Generated output combined data of ESMValTool processing..
- **archive** (*application/zip*) – The complete output of the ESMValTool processing as an zip archive.
- **success** (*string*) – True if the metric has been successfully calculated. If false please check the log files
- **recipe** (*text/plain*) – ESMValTool recipe used for processing.
- **log** (*text/plain*) – Log File of ESMValTool processing.
- **debug_log** (*text/plain*) – Debug Log File of ESMValTool processing.

References

- ESMValTool
- Documentation
- Estimated Calculation Time

get_outputs (*result*, *constraints*, *response*)

class c3s_magic_wps.processes.**HeatwavesColdwaves**
Bases: pywps.app.Process.Process

heatwaves_coldwaves Heatwave and coldwave duration (v1.0.0)

Metric showing the duration of heatwaves and coldwaves, to help understand potential changes in energy demand.

Parameters

- **model** (*{'None'}*) – Choose a model
- **experiment** (*{'None'}*) – Choose an experiment
- **ensemble** (*{'None'}*) – Choose an ensemble
- **start_historical** (*integer*) – Start historical of model data.
- **end_historical** (*integer*) – End historical of model data.
- **start_projection** (*integer*) – Start projection of model data.
- **end_projection** (*integer*) – End projection of model data.
- **quantile** (*float*) – Quantile defining the exceedance/non-exceedance threshold.
- **min_duration** (*integer*) – Minimum duration in days of a heatwave/coldwave event.
- **operator** (*{'exceedances', 'non-exceedances'}*) – Exceedance/non-exceedance of historic threshold.
- **season** (*{'summer', 'winter'}*) – Choose a season.

Returns

- **plot** (*image/png*) – Generated extreme spell duration tasmin plot.
- **data** (*application/zip*) – Extreme spell duration tasmin data.
- **archive** (*application/zip*) – The complete output of the ESMValTool processing as an zip archive.
- **success** (*string*) – True if the metric has been successfully calculated. If false please check the log files
- **recipe** (*text/plain*) – ESMValTool recipe used for processing.
- **log** (*text/plain*) – Log File of ESMValTool processing.
- **debug_log** (*text/plain*) – Debug Log File of ESMValTool processing.

References

- ESMValTool
- Documentation
- Media
- Estimated Calculation Time

get_outputs(*result*, *constraints*, *response*)

class c3s_magic_wps.processes.**HyInt**

Bases: pywps.app.Process.Process

hyint HyInt - Hydroclimatic intensity and extremes (v1.0.0)

HyInt hydroclimatic indices calculation and plotting.

Parameters

- **model** (*{'None'}*) – Choose a model
- **experiment** (*{'None'}*) – Choose an experiment
- **ensemble** (*{'None'}*) – Choose an ensemble
- **start_year** (*integer*) – Start year of model data.
- **end_year** (*integer*) – End year of model data.
- **ref_model** (*{'ACCESS1-0'}*) – Choose a reference model like ACCESS1-0.
- **indices** (*{'pa_norm', 'hyint', 'int_norm', 'r95_norm', 'wsl_norm', 'dsl_norm', 'int', 'dsl', 'wsl'}*) – Indices to be analysed and plotted.
- **regions** (*{'GL', 'GL60', 'TR', 'SA', 'AF', 'IN', 'EU', 'EA', 'AU'}*) – Regions for timeseries and maps.
- **norm_year_start** (*integer*) – First year of reference normalization period to be used for normalized indices.
- **norm_year_end** (*integer*) – Last year of reference normalization period to be used for normalized indices.

Returns

- **plot1** (*image/png*) – Single panel lon/lat map per individual index, multi-year mean
- **plot2** (*image/png*) – 3-panel lon/lat maps per individual index with comparison to reference dataset, multi-year mean
- **plot3** (*image/png*) – multipanel of indices of lon/lat maps with comparison to reference dataset, multi-year mean
- **plot12** (*image/png*) – multipanel of indices with timeseries over multiple regions
- **plot13** (*image/png*) – multipanel of indices with timeseries for multiple models
- **plot14** (*image/png*) – multipanel of indices with summary of trend coefficients over multiple regions
- **plot15** (*image/png*) – multipanel of indices with summary of trend coefficients for multiple models
- **archive** (*application/zip*) – The complete output of the ESMValTool processing as a zip archive.
- **success** (*string*) – True if the metric has been successfully calculated. If false please check the log files
- **recipe** (*text/plain*) – ESMValTool recipe used for processing.
- **log** (*text/plain*) – Log File of ESMValTool processing.
- **debug_log** (*text/plain*) – Debug Log File of ESMValTool processing.

References

- ESMValTool
- Documentation
- Estimated Calculation Time

get_outputs (*models*, *result*, *response*)

class c3s_magic_wps.processes.**ModesVariability**
Bases: pywps.app.Process.Process

modes_of_variability Modes of variability (v1.0.0)

Diagnostics showing the RMSE between the observed and modelled patterns of variability obtained through classification and their relative relative bias (percentage) in the frequency of occurrence and the persistence of each mode.

Parameters

- **model** (*{'None'}*) – Choose a model
- **experiment** (*{'None'}*) – Choose an experiment
- **ensemble** (*{'None'}*) – Choose an ensemble
- **start_historical** (*integer*) – Start historical of model data.
- **end_historical** (*integer*) – End historical of model data.
- **start_projection** (*integer*) – Start projection of model data.
- **end_projection** (*integer*) – End projection of model data.
- **plot_type** (*{'polar', 'rectangular'}*) – Plot type.
- **ncenters** (*{'3', '4', '5'}*) – Choose a number of cluster centers.
- **detrend_order** (*{'2', '1'}*) – Choose a order of detrend.
- **cluster_method** (*{'kmeans', 'hclust'}*) – Choose a clustering method.
- **eofs** (*{'True', 'False'}*) – Calculate EOFs?
- **frequency** (*{'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL', 'AUG', 'SEP', 'OCT', ..}*) – Choose a frequency like JAN.

Returns

- **table_psl_plot** (*image/png*) – Generated Table_psl plot of ESMValTool processing.
- **psl_predicted_regimes_plot** (*image/png*) – Generated psl_predicted_regimes plot of ESMValTool processing.
- **psl_observed_regimes_plot** (*image/png*) – Generated psl_observed_regimes plot of ESMValTool processing.
- **rmse** (*application/x-netcdf*) – Generated output data of ESMValTool processing.
- **exp** (*application/x-netcdf*) – Generated output data of ESMValTool processing.
- **obs** (*application/x-netcdf*) – Generated output data of ESMValTool processing.
- **archive** (*application/zip*) – The complete output of the ESMValTool processing as an zip archive.

- **success** (*string*) – True if the metric has been successfully calculated. If false please check the log files
- **recipe** (*text/plain*) – ESMValTool recipe used for processing.
- **log** (*text/plain*) – Log File of ESMValTool processing.
- **debug_log** (*text/plain*) – Debug Log File of ESMValTool processing.

References

- ESMValTool
- Documentation
- Estimated Calculation Time

get_outputs (*result*, *response*)

class c3s_magic_wps.processes.**MultimodelProducts**

Bases: pywps.app.Process.Process

multimodel_products Generic multi-model products (v1.0.0)

For the 'generic multi-model diagnostic' the ensemble mean anomaly, and the ensemble variance and agreement are calculated. The results are shown as maps and time series.

Parameters

- **model** (*{'None'}*) – Choose a model
- **experiment** (*{'None'}*) – Choose an experiment
- **ensemble** (*{'None'}*) – Choose an ensemble
- **start_historical** (*integer*) – Start historical of model data.
- **end_historical** (*integer*) – End historical of model data.
- **start_projection** (*integer*) – Start projection of model data.
- **end_projection** (*integer*) – End projection of model data.
- **moninf** (*{'1', '2', '3', '4', '5', '6', '7', '8', '9', '10', . .}*) – First month of the seasonal mean period to be computed, if none the monthly anomalies will be computed.
- **monsup** (*{'1', '2', '3', '4', '5', '6', '7', '8', '9', '10', . .}*) – Last month of the seasonal mean period to be computed, if none the monthly anomalies will be computed.
- **agreement_threshold** (*integer*) – Threshold in percent for the minimum agreement between models on the sign of the multi-model mean anomaly for the stipling to be plotted.
- **running_mean** (*integer*) – Length of the window for which the running mean is computed.
- **time_series_plot** (*{'single', 'maxmin'}*) – Either single or maxmin (plot the individual or the mean with shading between the max and min).

Returns

- **tas_plot** (*image/png*) – Generated tas plot of ESMValTool processing.

- **area_plot** (*image/png*) – Generated Area plot of ESMValTool processing.
- **data** (*application/x-netcdf*) – Generated output data of ESMValTool processing.
- **archive** (*application/zip*) – The complete output of the ESMValTool processing as an zip archive.
- **success** (*string*) – True if the metric has been successfully calculated. If false please check the log files
- **recipe** (*text/plain*) – ESMValTool recipe used for processing.
- **log** (*text/plain*) – Log File of ESMValTool processing.
- **debug_log** (*text/plain*) – Debug Log File of ESMValTool processing.

References

- ESMValTool
- Documentation
- Media
- Model Selection
- Estimated Calculation Time

get_outputs (*result*, *response*)

class c3s_magic_wps.processes.**PreprocessExample**

Bases: pywps.app.Process.Process

preproc Preprocessing Demo (v1.0.0)

The ESMValTool climate data pre-processor can be used to perform all types of climate data pre-processing needed before indices or diagnostics can be calculated. It is a base component for many other diagnostics and metrics shown on this portal. It can be applied to tailor the climate model data to the need of the user for its own calculations.

Parameters

- **model** (*{'None'}*) – Choose a model
- **experiment** (*{'None'}*) – Choose an experiment
- **ensemble** (*{'None'}*) – Choose an ensemble
- **start_year** (*integer*) – Start year of model data.
- **end_year** (*integer*) – End year of model data.
- **extract_levels** (*float*) – Choose an extraction level for the preprocessor.

Returns

- **multi_model_mean_ta_plot** (*image/png*) – Generated multi_model_mean_ta plot of ESMValTool processing.
- **multi_model_median_ta_plot** (*image/png*) – Generated multi_model_median_ta plot of ESMValTool processing.
- **model1_mean_ta_plot** (*image/png*) – Generated model1_mean_ta plot of ESMValTool processing.

- **model2_mean_ta_plot** (*image/png*) – Generated model2_mean_ta plot of ESMValTool processing.
- **reference_model_mean_ta_plot** (*image/png*) – Generated reference_model_mean_ta plot of ESMValTool processing.
- **model1_mean_pr_plot** (*image/png*) – Generated model1_mean_pr plot of ESMValTool processing.
- **model2_mean_pr_plot** (*image/png*) – Generated model2_mean_pr plot of ESMValTool processing.
- **reference_model_mean_pr_plot** (*image/png*) – Generated reference_model_mean_pr plot of ESMValTool processing.
- **multi_model_mean_ta_data** (*application/x-netcdf*) – Generated output multi_model_mean_ta data of ESMValTool processing..
- **multi_model_median_ta_data** (*application/x-netcdf*) – Generated output multi_model_median_ta data of ESMValTool processing..
- **model1_mean_ta_data** (*application/x-netcdf*) – Generated output model1_mean_ta data of ESMValTool processing..
- **model2_mean_ta_data** (*application/x-netcdf*) – Generated output model2_mean_ta data of ESMValTool processing..
- **reference_model_mean_ta_data** (*application/x-netcdf*) – Generated output reference_model_mean_ta data of ESMValTool processing..
- **model1_mean_pr_data** (*application/x-netcdf*) – Generated output model1_mean_pr data of ESMValTool processing..
- **model2_mean_pr_data** (*application/x-netcdf*) – Generated output model2_mean_pr data of ESMValTool processing..
- **reference_model_mean_pr_data** (*application/x-netcdf*) – Generated output reference_model_mean_pr data of ESMValTool processing..
- **archive** (*application/zip*) – The complete output of the ESMValTool processing as an zip archive.
- **success** (*string*) – True if the metric has been successfully calculated. If false please check the log files
- **recipe** (*text/plain*) – ESMValTool recipe used for processing.
- **log** (*text/plain*) – Log File of ESMValTool processing.
- **debug_log** (*text/plain*) – Debug Log File of ESMValTool processing.

References

- ESMValTool
- Documentation
- Model Selection
- Estimated Calculation Time

get_outputs (*result*, *constraints*, *response*)

class c3s_magic_wps.processes.**QuantileBias**
Bases: pywps.app.Process.Process

quantile_bias Quantile Bias (v1.0.0)

Diagnostic showing the quantile bias between models and a reference dataset.

Parameters

- **model** (*{'None'}*) – Choose a model
- **experiment** (*{'None'}*) – Choose an experiment
- **ensemble** (*{'None'}*) – Choose an ensemble
- **start_year** (*integer*) – Start year of model data.
- **end_year** (*integer*) – End year of model data.
- **ref_dataset** (*{'GPCP-SG'}*) – Choose a reference dataset like GPCP-SG.
- **perc_lev** (*integer*) – Quantile in percentage (%).

Returns

- **model** (*application/x-netcdf*) – Generated output data of ESMValTool processing.
- **archive** (*application/zip*) – The complete output of the ESMValTool processing as a zip archive.
- **success** (*string*) – True if the metric has been successfully calculated. If false please check the log files
- **recipe** (*text/plain*) – ESMValTool recipe used for processing.
- **log** (*text/plain*) – Log File of ESMValTool processing.
- **debug_log** (*text/plain*) – Debug Log File of ESMValTool processing.

References

- ESMValTool
- Documentation
- Estimated Calculation Time

get_outputs(*model*, *result*, *response*)

class c3s_magic_wps.processes.**RainFARM**
Bases: pywps.app.Process.Process

rainfarm RainFARM stochastic downscaling (v1.0.0)

Precipitation extremes and small-scale variability are essential drivers in many climate change impact studies. However, the spatial resolution currently achieved by global and regional climate models is still insufficient to correctly identify the fine structure of precipitation intensity fields. In the absence of a proper physically based representation, this scale gap can be at least temporarily bridged by adopting a stochastic rainfall downscaling technique (Rebora et al, 2006). With this aim, the Rainfall Filtered Autoregressive Model (RainFARM)was developed to apply the stochastic precipitation downscaling method to climate models. The selected region needs to have equal and even number of longitude (in any case it is cut) and latitude grid points (e.g., 2x2, 4x4, . . .). Warning: downcaling can reach very high resolution, so select a limited area.

Parameters

- **model** (*{'None'}*) – Choose a model
- **experiment** (*{'None'}*) – Choose an experiment
- **ensemble** (*{'None'}*) – Choose an ensemble
- **start_year** (*integer*) – Start year of model data.
- **end_year** (*integer*) – End year of model data.
- **start_longitude** (*integer*) – Minimum longitude.
- **end_longitude** (*integer*) – Maximum longitude.
- **start_latitude** (*integer*) – Minimum latitude.
- **end_latitude** (*integer*) – Maximum latitude.
- **target_grid** (*string*) – Target grid in degrees (e.g. 1x1) can also be the name of one of the datasets to use the grid from that dataset.
- **scheme** (*{'linear', 'nearest', 'area_weighted', 'unstructured_nearest'}*) – Regridding scheme to be used.
- **slope** (*float*) – Spatial spectral slope (set to 0 to compute fromlarge scales).
- **nens** (*integer*) – Number of ensemble members to be calculated.
- **nf** (*integer*) – Number of subdivisions for downscaling (e.g. 8 will produce output fields with linear resolution increased by a factor 8).
- **conserv_glob** (*{'true', 'false'}*) – Conserve precipitation over full domain?
- **conserv_smooth** (*{'true', 'false'}*) – Conserve precipitation using convolution (if neither is chosen box conservation is used)?

Returns

- **archive** (*application/zip*) – The complete output of the ESMValTool processing as an zip archive.
- **success** (*string*) – True if the metric has been successfully calculated. If false please check the log files
- **recipe** (*text/plain*) – ESMValTool recipe used for processing.
- **log** (*text/plain*) – Log File of ESMValTool processing.
- **debug_log** (*text/plain*) – Debug Log File of ESMValTool processing.

References

- ESMValTool
- Documentation
- Estimated Calculation Time

get_outputs(*result*, *response*)

class c3s_magic_wps.processes.**ShapeSelect**

Bases: pywps.app.Process.Process

shapefile_selection Shapefile selection (v1.0.0)

Metric showing selected gridded data within a user defined polygon shapefile and outputting as NetCDF or csv file.

Parameters

- **model** (*{'None'}*) – Choose a model
- **experiment** (*{'None'}*) – Choose an experiment
- **ensemble** (*{'None'}*) – Choose an ensemble
- **start_year** (*integer*) – Start year of model data.
- **end_year** (*integer*) – End year of model data.
- **shape** (*{'MotalaStrom', 'Elbe', 'multicatchment', 'testfile', 'Thames'}*) – Shape of the area
- **weighting_method** (*{'mean_inside', 'representative'}*) – The preferred weighting method: mean_inside - mean of all grid points inside polygon or representative - one point inside or close to the polygon is used to represent the complete area.

Returns

- **data** (*application/x-netcdf*) – Generated NetCDF file with precipitation for the selected area.
- **xlsx_data** (*application/vnd.ms-excel*) – Generated excel file with precipitation for the selected area
- **archive** (*application/zip*) – The complete output of the ESMValTool processing as an zip archive.
- **success** (*string*) – True if the metric has been successfully calculated. If false please check the log files
- **recipe** (*text/plain*) – ESMValTool recipe used for processing.
- **log** (*text/plain*) – Log File of ESMValTool processing.
- **debug_log** (*text/plain*) – Debug Log File of ESMValTool processing.

References

- ESMValTool
- Documentation
- Estimated Calculation Time

get_outputs (*result*, *response*)

class c3s_magic_wps.processes.**Sleep**

Bases: pywps.app.Process.Process

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** (*float*) – Delay between every update

Returns **sleep_output** – Sleep Output

Return type string

References

- PyWPS Demo

class c3s_magic_wps.processes.**Teleconnections**

Bases: pywps.app.Process.Process

teleconnections Teleconnection indices (v1.0.0)

Diagnostic providing teleconnection indices (Z500 empirical orthogonal functions)

Parameters

- **model** (*{'None'}*) – Choose a model
- **experiment** (*{'None'}*) – Choose an experiment
- **ensemble** (*{'None'}*) – Choose an ensemble
- **start_year** (*integer*) – Start year of model data.
- **end_year** (*integer*) – End year of model data.
- **ref_model** (*{'ERA-Interim'}*) – Choose a reference model like ERA-Interim.
- **season** (*{'DJF', 'MAM', 'JJA', 'SON', 'ALL'}*) – Choose a season like DJF.
- **teles** (*{'NAO', 'AO', 'PNA'}*) – Choose an EOF like NAO.

Returns

- **eof1_plot** (*image/png*) – Generated EOF1 plot of ESMValTool processing.
- **eof2_plot** (*image/png*) – Generated EOF2 plot of ESMValTool processing.
- **eof3_plot** (*image/png*) – Generated EOF3 plot of ESMValTool processing.
- **eof4_plot** (*image/png*) – Generated EOF4 plot of ESMValTool processing.
- **data** (*application/x-netcdf*) – Generated output data of ESMValTool processing.
- **archive** (*application/zip*) – The complete output of the ESMValTool processing as an zip archive.
- **success** (*string*) – True if the metric has been successfully calculated. If false please check the log files
- **recipe** (*text/plain*) – ESMValTool recipe used for processing.
- **log** (*text/plain*) – Log File of ESMValTool processing.
- **debug_log** (*text/plain*) – Debug Log File of ESMValTool processing.

References

- ESMValTool
- Documentation
- Estimated Calculation Time

get_outputs(*result*, *subdir*, *response*)

class c3s_magic_wps.processes.**Toymodel**
Bases: pywps.app.Process.Process

toymodel Toymodel (v1.0.0)

The goal of this diagnostic is to simulate single-model ensembles from an observational dataset to investigate the effect of observational uncertainty.

Parameters

- **model** (*{'None'}*) – Choose a model
- **experiment** (*{'None'}*) – Choose an experiment
- **ensemble** (*{'None'}*) – Choose an ensemble
- **start_year** (*integer*) – Start year of model data.
- **end_year** (*integer*) – End year of model data.
- **variable** (*{'psl', 'tas'}*) – Select the variable to simulate.
- **start_longitude** (*integer*) – Minimum longitude.
- **end_longitude** (*integer*) – Maximum longitude.
- **start_latitude** (*integer*) – Minimum latitude.
- **end_latitude** (*integer*) – Maximum latitude.
- **beta** (*float*) – User defined underdispersion (beta >= 0).
- **number_of_members** (*integer*) – Number of members to be generated.

Returns

- **plot** (*image/jpeg*) – Generated synthetic model plt.
- **model** (*application/x-netcdf*) – Generated synthetic model.
- **archive** (*application/zip*) – The complete output of the ESMValTool processing as an zip archive.
- **success** (*string*) – True if the metric has been successfully calculated. If false please check the log files
- **recipe** (*text/plain*) – ESMValTool recipe used for processing.
- **log** (*text/plain*) – Log File of ESMValTool processing.
- **debug_log** (*text/plain*) – Debug Log File of ESMValTool processing.

References

- ESMValTool
- Documentation

get_outputs(*result*, *response*)

class c3s_magic_wps.processes.**WeatherRegimes**
Bases: pywps.app.Process.Process

weather_regimes Weather regimes (v1.0.0)

Diagnostic providing North-Atlantic Weather Regimes

Parameters

- **model** (*{'None'}*) – Choose a model
- **experiment** (*{'None'}*) – Choose an experiment
- **ensemble** (*{'None'}*) – Choose an ensemble
- **start_year** (*integer*) – Start year of model data.
- **end_year** (*integer*) – End year of model data.
- **ref_model** (*{'ERA-Interim'}*) – Choose a reference model like ERA-Interim.

Returns

- **regime1_plot** (*image/png*) – Generated Regime1 plot of ESMValTool processing.
- **regime2_plot** (*image/png*) – Generated Regime2 plot of ESMValTool processing.
- **regime3_plot** (*image/png*) – Generated Regime3 plot of ESMValTool processing.
- **regime4_plot** (*image/png*) – Generated Regime4 plot of ESMValTool processing.
- **data** (*application/x-netcdf*) – Generated output data of ESMValTool processing.
- **archive** (*application/zip*) – The complete output of the ESMValTool processing as an zip archive.
- **success** (*string*) – True if the metric has been successfully calculated. If false please check the log files
- **recipe** (*text/plain*) – ESMValTool recipe used for processing.
- **log** (*text/plain*) – Log File of ESMValTool processing.
- **debug_log** (*text/plain*) – Debug Log File of ESMValTool processing.

References

- ESMValTool
- Documentation
- Estimated Calculation Time

get_outputs(*result*, *subdir*, *response*)

class c3s_magic_wps.processes.**ZMNAM**

Bases: pywps.app.Process.Process

zmnam Stratosphere-troposphere coupling and annular modes indices (ZMNAM) (v1.0.0)

Stratosphere-troposphere coupling and annular modes indices (ZMNAM)

Parameters

- **model** (*{'None'}*) – Choose a model
- **experiment** (*{'None'}*) – Choose an experiment
- **ensemble** (*{'None'}*) – Choose an ensemble
- **start_year** (*integer*) – Start year of model data.
- **end_year** (*integer*) – End year of model data.

Returns

- **5000pa_mo_reg_plot** (*image/png*) – Generated 5000Pa_mo_reg plot of ESMValTool processing.
- **25000pa_mo_reg_plot** (*image/png*) – Generated 25000Pa_mo_reg plot of ESMValTool processing.
- **50000pa_mo_reg_plot** (*image/png*) – Generated 50000Pa_mo_reg plot of ESMValTool processing.
- **100000pa_mo_reg_plot** (*image/png*) – Generated 100000Pa_mo_reg plot of ESMValTool processing.
- **5000pa_da_pdf_plot** (*image/png*) – Generated 5000Pa_da_pdf plot of ESMValTool processing.
- **25000pa_da_pdf_plot** (*image/png*) – Generated 25000Pa_da_pdf plot of ESMValTool processing.
- **50000pa_da_pdf_plot** (*image/png*) – Generated 50000Pa_da_pdf plot of ESMValTool processing.
- **100000pa_da_pdf_plot** (*image/png*) – Generated 100000Pa_da_pdf plot of ESMValTool processing.
- **5000pa_mo_ts_plot** (*image/png*) – Generated 5000Pa_mo_ts plot of ESMValTool processing.
- **25000pa_mo_ts_plot** (*image/png*) – Generated 25000Pa_mo_ts plot of ESMValTool processing.
- **50000pa_mo_ts_plot** (*image/png*) – Generated 50000Pa_mo_ts plot of ESMValTool processing.
- **100000pa_mo_ts_plot** (*image/png*) – Generated 100000Pa_mo_ts plot of ESMValTool processing.
- **regr_map** (*application/x-netcdf*) – Generated output data of ESMValTool processing.
- **eofs** (*application/x-netcdf*) – Generated output data of ESMValTool processing.
- **pc_mo** (*application/x-netcdf*) – Generated output data of ESMValTool processing.
- **pc_da** (*application/x-netcdf*) – Generated output data of ESMValTool processing.
- **archive** (*application/zip*) – The complete output of the ESMValTool processing as an zip archive.
- **success** (*string*) – True if the metric has been successfully calculated. If false please check the log files
- **recipe** (*text/plain*) – ESMValTool recipe used for processing.
- **log** (*text/plain*) – Log File of ESMValTool processing.
- **debug_log** (*text/plain*) – Debug Log File of ESMValTool processing.

References

- ESMValTool
- Documentation
- Estimated Calculation Time

get_outputs(*result*, *response*)

2.4 CHANGELOG

2.4.1 1.0.0rc1 (2019-06-06)

- First release candidate of stable release of c3s-magic-wps. A Web Processing Service for Climate Data Analysis in the MAGIC project. The software in this WPS powers the processes behind the C3S-MAGIC portal. The processes are run on CP4CDS infrastructure.

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Python Module Index

C

Index

P

Q

R

S

T

W

Z