
Earth Diagnostics Documentation

Release 3.0.0b56

BSC-CNS Earth Sciences Department

Aug 25, 2017

1	Tutorial	1
1.1	Installation	1
1.2	Creating a config file	1
1.3	Prepare the run script	2
2	Configuration file options	3
2.1	DIAGNOSTICS	3
2.1.1	Mandatory configurations	3
2.1.2	Optional configurations	3
2.2	EXPERIMENT	4
2.3	CMOR	4
2.3.1	Cmorization options	4
2.3.2	Metadata options	5
2.4	THREDDS	6
2.5	ALIAS	6
3	Diagnostic list	7
3.1	General	7
3.1.1	att	7
3.1.2	dailymean	7
3.1.3	module	8
3.1.4	monmean	8
3.1.5	relink	8
3.1.6	relinkall	9
3.1.7	rewrite:	9
3.1.8	scale	9
3.1.9	simdim	10
3.1.10	yearlymean	10
3.2	Ocean	10
3.2.1	areamoc	10
3.2.2	averagesection	11
3.2.3	convectionsites	11
3.2.4	cutsection	11
3.2.5	gyres	11
3.2.6	heatcontent	12
3.2.7	heatcontentlayer	12
3.2.8	interpolate	12
3.2.9	interpolateCDO	13
3.2.10	maskland	13
3.2.11	maxmoc	13
3.2.12	mixedlayerheatcontent	14

3.2.13	mixedlayersaltcontent	14
3.2.14	moc	14
3.2.15	mxl	14
3.2.16	psi	14
3.2.17	regmean	15
3.2.18	rotate	15
3.2.19	siasiesiv	15
3.2.20	vgrad	16
3.2.21	verticalmean	16
3.2.22	verticalmeanmeters	16
3.3	Statistics	16
3.3.1	climpercent	16
3.3.2	monpercent	17
4	Tips and tricks	19
4.1	Working with ORCA1	19
4.2	Configuring core usage	19
4.3	NEMO files	19
5	What to do if you have an error	21
6	Developer’s guide	23
6.1	Developing a diagnostic	23
7	Frequently Asked Questions	25
8	Module documentation	27
8.1	earthdiagnostics	28
8.1.1	earthdiagnostics.box	28
8.1.2	earthdiagnostics.cdftools	28
8.1.3	earthdiagnostics.cmorizer	28
8.1.4	earthdiagnostics.cmormanager	28
8.1.5	earthdiagnostics.config	28
8.1.6	earthdiagnostics.constants	28
8.1.7	earthdiagnostics.datamanager	28
8.1.8	earthdiagnostics.diagnostic	28
8.1.9	earthdiagnostics.earthdiags	28
8.1.10	earthdiagnostics.parser	28
8.1.11	earthdiagnostics.utils	28
8.1.12	earthdiagnostics.variable	28
8.2	earthdiagnostics.general	28
8.2.1	earthdiagnostics.general.attribute	28
8.2.2	earthdiagnostics.general.dailymean	28
8.2.3	earthdiagnostics.general.module	28
8.2.4	earthdiagnostics.general.monthlymean	28
8.2.5	earthdiagnostics.general.relink	28
8.2.6	earthdiagnostics.general.relinkall	28
8.2.7	earthdiagnostics.general.rewrite	28
8.2.8	earthdiagnostics.general.scale	28
8.2.9	earthdiagnostics.general.simplify_dimensions	28
8.2.10	earthdiagnostics.general.yearlymean	28
8.3	earthdiagnostics.ocean	28
8.3.1	earthdiagnostics.ocean.areamoc	28
8.3.2	earthdiagnostics.ocean.averagesection	28
8.3.3	earthdiagnostics.ocean.convectionsites	28

8.3.4	earthdiagnostics.ocean.cutsection	28
8.3.5	earthdiagnostics.ocean.gyres	28
8.3.6	earthdiagnostics.ocean.heatcontent	28
8.3.7	earthdiagnostics.ocean.heatcontentlayer	28
8.3.8	earthdiagnostics.ocean.interpolate	28
8.3.9	earthdiagnostics.ocean.interpolatecd	28
8.3.10	earthdiagnostics.ocean.maskland	28
8.3.11	earthdiagnostics.ocean.maxmoc	28
8.3.12	earthdiagnostics.ocean.mixedlayerheatcontent	28
8.3.13	earthdiagnostics.ocean.mixedlayersaltcontent	28
8.3.14	earthdiagnostics.ocean.moc	28
8.3.15	earthdiagnostics.ocean.mxl	28
8.3.16	earthdiagnostics.ocean.psi	28
8.3.17	earthdiagnostics.ocean.rotation	28
8.3.18	earthdiagnostics.ocean.siasiesiv	28
8.3.19	earthdiagnostics.ocean.verticalgradient	28
8.3.20	earthdiagnostics.ocean.verticalmean	28
8.3.21	earthdiagnostics.ocean.verticalmeanmeters	28
8.4	earthdiagnostics.statistics	28
8.4.1	earthdiagnostics.statistics.climatologicalpercentile	28
8.4.2	earthdiagnostics.statistics.monthlypercentile	28

TUTORIAL

So, you are planning to use the Earth Diagnostics? You don't know how to use them? This is the place to go. From now on this tutorial will guide you through all the process from installation to running.

Hint: If you have any problem with this tutorial, please report it to <javier.vegas@bsc.es> so it can be corrected. A lot of people will benefit from it.

1.1 Installation

If you have access to the BSC-ES machines, you don't need to install it. Just use the available module:

In case that you need a custom installation for development or can not use the BSC-ES machines, install it from BSC-ES GitLab repository:

```
pip install git+https://earth.bsc.es/gitlab/es/ocean_diagnostics.git
```

You will also need

- CDO version 1.7.2 (other versions could work, but this is the one we use)
- NCO version 4.5.4 or newer
- Python 2.7 or newer (but no 3.x) with bscearth.utils, CDO and NCO packages, among others.
- Access to CDFTOOLS_3.0 executables for BSC-ES. The source code is available on Github (<https://github.com/jvegasbsc/CDFTOOLS>) and it can be compiled with CMake

1.2 Creating a config file

Go to the folder where you installed the EarthDiagnostics. You will see a folder called earthdiagnostics, and, inside it, a diags.conf file that can be used as a model for your config file. Create a copy of it wherever it suites you.

Now open your brand new copy with your preferred text editor. The file contains commentaries explaining each one of its options, so read it carefully and edit whatever you need. Don't worry about DIAGS option, we will talk about it next.

After this, you need to choose the diagnostics you want to run. For a simple test, it's recommended to use the monmean diagnostic to compute monthly means from daily data. We recommend it because it can be used with any variable, the user has to provide parameters but they are quite intuitive and it's relatively fast to compute. If your experiment does not have daily data, you can use any other diagnostic. Check next section for a list of available diagnostics and choose whichever suits you better. From now on, we will assume that you are going to run the monmean diagnostic.

Hint: For old Ocean Diagnostics users: you can use most of the old names as aliases to launch one or multiple diagnostics. Check the ALIAS section on the diags.conf to see which ones are available.

First, choose a variable that has daily data. Then replace the DIAGS option with the next one where \$VARIABLE represents the variable's name and \$DOMAIN its domain (atmos, ocean, seaice, landice...)

```
DIAGS = monmean, $VARIABLE, $DOMAIN
```

1.3 Prepare the run script

Once you have configured your experiment you can execute any diagnostic with the provided launch_diags.sh script. Create a copy and change the variables PATH_TO_CONF_FILE and PATH_TO_DIAGNOSTICS so they point to your conf file and installation folder.

Now, execute the script (or submit it to bsclogin01, it has the correct header) and... that's it! You will find your results directly on the storage and a folder for the temp files in the scratch named after the EXPID.

CONFIGURATION FILE OPTIONS

This section contains the list and explanation about all the options that are available on the configuration file. Use it as a reference while preparing your configuration file. Each subsection will refer to the matching section from the config file. Those subsections explanation may be divided itself for the shake of clarity but this further divisions have nothing to do with the config file syntax itself.

2.1 DIAGNOSTICS

This section contains the general configuration for the diagnostics. The explanation has been divided in two subsections: the first one will cover all the mandatory options that you must specify in every configuration, while the second will cover all the optional configurations.

2.1.1 Mandatory configurations

- **SCRATCH_DIR:** Temporary folder for the calculations. Final results will never be stored here.
- **DATA_DIR:** ‘.’ separated list of folders to look for data in. It will look for file in the path \$DATA_FOLDER/\$EXPID and \$DATA_FOLDER/\$DATA_TYPE/\$MODEL/\$EXPID
- **CON_FILES:** Folder containing mask and mesh files for the dataset.
- **FREQUENCY:** Default data frequency to be used by the diagnostics. Some diagnostics can override this configuration or even ignore it completely.
- **DIAGS:** List of diagnostic to run, in the order you want them to run

2.1.2 Optional configurations

- **SCRATCH_MASKS** Common scratch folder for the ocean masks. This is useful to avoid replicating them for each run at the fat nodes. By default is ‘/scratch/Earth/ocean_masks’
- **RESTORE_MESHES** By default, Earth Diagnostics only copies the mask files if they are not present in the scratch folder. If this option is set to true, Earth Diagnostics will copy them regardless of existence. Default is False.
- **DATA_ADAPTOR** This is used to choose the mechanism for storing and retrieving data. Options are CMOR (for our own experiments) or THREDDS (for anything else). Default value is CMOR
- **DATA_TYPE** Type of the dataset to use. It can be exp, obs or recon. Default is exp.
- **DATA_CONVENTION** Convention to use for file paths and names and variable naming among other things. Can be SPECS, PRIMAVERA or CMIP6. Default is SPECS.

- **CDFTOOLS_PATH** Path to the folder containing CDFTOOLS executables. By default is empty, so CDFTOOLS binaries must be added to the system path.
- **MAX_CORES** Maximum number of cores to use. By default the diagnostics will use all cores available to them. It is not necessary when launching through a scheduler, as Earthdiagnostics can detect how many cores the scheduler has allocated to it.

2.2 EXPERIMENT

This sections contains options related to the experiment's definition or configuration.

- **MODEL** Name of the model used for the experiment.
- **MODEL_VERSION** Model version. Used to get the correct mask and mesh files
- **ATMOS_Timestep** Time between outputs from the atmosphere. This is not the model simulation timestep! Default is 6.
- **OCEAN_Timestep** Time between outputs from the ocean. This is not the model simulation timestep! Default is 6.
- **ATMOS_GRID** Atmospheric grid definition. Will be used as a default target for interpolation diagnostics.
- **INSTITUTE** Institute that made the experiment, observation or reconstruction
- **EXPID** Unique identifier for the experiment
- **NAME** Experiment's name. By default it is the EXPID.
- **STARTDATES** Startdates to run as a space separated list
- **MEMBER** Members to run as a space separated list. You can just provide the number or also add the prefix
- **MEMBER_DIGITS** Number of minimum digits to compose the member name. By default it is 1. For example, for member 1 member name will be fc1 if MEMBER_DIGITS is 1 or fc01 if MEMBER_DIGITS is 2
- **MEMBER_PREFIX** Prefix to use for the member names. By default is 'fc'
- **MEMBER_COUNT_START** Number corresponding to the first member. For example, if your first member is 'fc1', it should be 1. If it is 'fc0', it should be 0. By default is 0
- **CHUNK_SIZE** Length of the chunks in months
- **CHUNKS** Number of chunks to run
- **CALENDAR** Calendar to use for date calculation. All calendars supported by Autosubmit are available. Default is 'standard'

2.3 CMOR

In this section, you can control how will work the cmorization process. All options belonging to this section are optional.

2.3.1 Cmorization options

This options control when and which varibales will be cmorized.

- **FORCE** If True, launches the cmorization, regardless of existence of the extracted files or the package containing the online-cmorized ones. If False, only the non-present chunks will be cmorized. Default value is False
- **FORCE_UNTAR** Unpacks the online-cmorized files regardless of existence of extracted files. If 'FORCE' is True, this parameter has no effect. If False, only the non-present chunks will be unpacked. Default value is False.
- **FILTER_FILES** Only cmorize original files containing any of the given strings. This is a space separated list. Default is the empty string.
- **OCEAN_FILES** Boolean flag to activate or no NEMO files cmorization. Default is True.
- **ATMOSPHERE_FILES** Boolean flag to activate or no IFS files cmorization. Default is True.
- **USE_GRIB** Boolean flag to activate or no GRIB files cmorization for the atmosphere. If activated and no GRIB files are present, it will cmorize using the MMA files instead (as if it was set to False). Default is True.
- **CHUNKS** Space separated list of chunks to be cmorized. If not provided, all chunks are cmorized
- **VARIABLE_LIST** Space separated list of variables to cmorize. Variables must be specified as domain:var_name. If no one is specified, all the variables will be cmorized

Grib variables extraction

These three options are used to configure the variables to be CMORized from the grib atmospheric files. They must be specified using the IFS code in a list separated by comma.

You can also specify the levels to extract using one of the the following syntaxes:

- **VARIABLE_CODE**
- **VARIABLE_CODE:LEVEL**,
- **VARIABLE_CODE:LEVEL_1-LEVEL_2-...-LEVEL_N**
- **VARIABLE_CODE:MIN_LEVEL:MAX_LEVEL:STEP**

Some examples to clarify it further: * Variable with code 129 at level 30000: 129:30000 * Variable with code 129 at levels 30000, 40000 and 60000: 129:30000-40000-60000 * Variable with code 129 at levels between 30000 and 600000 with 10000 intervals:

129:30000:60000:10000 equivalent to 129:30000-40000-50000-60000

- **ATMOS_HOURLY_VARS** Configuration of variables to be extracted in an hourly basis
- **ATMOS_DAILY_VARS** Configuration of variables to be extracted in a daily basis
- **ATMOS_MONTHLY_VARS** Configuration of variables to be extracted in a monthly basis

2.3.2 Metadata options

All the options in this subsection will serve just to add the given values to the homonymous attributes in the cmorized files.

- **ASSOCIATED_EXPERIMENT** Default value is 'to be filled'
- **ASSOCIATED_MODEL** Default value is 'to be filled'
- **INITIALIZATION_DESCRIPTION** Default value is 'to be filled'
- **INITIALIZATION_METHOD** Default value is '1'

- **PHYSICS_DESCRIPTION** Default value is ‘to be filled’
- **PHYSICS_VERSION** Default value is ‘1’
- **SOURCE** Default value is ‘to be filled’

2.4 THREDDS

For now, there is only one option for the THREDDS server configuration.

- **SERVER_URL** THREDDS server URL

2.5 ALIAS

This config file section is different from all the others because it does not contain a set of configurations. Instead, in this section the user can define a set of aliases to be able to launch its most used configurations with ease. To do this, the user must add an option with named after the desired alias and assign to it the configuration or configurations to launch when this ALIAS is invoked. See the next example:

```
ALIAS_NAME = diag,opt1,opt2 diag,opt1new,opt2
```

In this case, the user has defined a new alias ‘ALIAS’ that can be used launch two times the diagnostic ‘diag’, the first with the options ‘opt1’ and ‘opt2’ and the second replacing ‘opt1’ with ‘opt1new’.

In this example, configuring the DIAGS as

```
DIAGS = ALIAS_NAME
```

will be identical to

```
DIAGS = diag,opt1,opt2 diag,opt1new,opt2
```

```
# coding=utf-8
```

DIAGNOSTIC LIST

In this section you have a list of the available diagnostics, with a small description of each one and a link to the full documentation. To see what options are available for each diagnostic, see `generate_jobs` documentation.

Remember that diagnostics are specified separated by spaces while options are given separated by commas:

```
DIAGS = diag1 diag2,option1,option2 diag3
```

3.1 General

The diagnostics from this section are of general use and can be used with any variable you may have. Most of them are meant to help you to solve usual issues that you may have with the data: incorrect metadata, scaled up or down variables, links missing. This section also contains the diagnostic used to calculate the monthly means.

3.1.1 att

Writes a global attributte to all the netCDF files for a given variable. See `Attribute`

Options:

1. **Variable:** Variable name
2. **Domain:** Variable domain
3. **Attributte name:** Attributte to write
4. **Attribute value:** Attribute's new value. Replace ',' with '&,' and ' ' with '&.' to avoid parsing errors when processing the diags
5. **Grid = "":** Variable grid. Only required in case that you want to use interpolated data.

3.1.2 dailymean

Calculates the daily mean for a given variable. See `DailyMean`

Warning: This diagnostic does not use the frequency configuration from the config file. You must specify the original frequency when calling it.

Options:

1. **Variable:** Variable name
2. **Domain:** Variable domain
3. **Original frequency:** Original frequency to use
4. **Grid = “”:** Variable grid. Only required in case that you want to use interpolated data.

3.1.3 module

Calculates the module for two given variables and stores the result in a third. See `Module`

Options:

1. **Domain:** Variables domain
2. **Variable U:** Variable U name
3. **Variable V:** Variable V name
4. **Variable Module:** Variable module name
5. **Grid = “”:** Variable grids. Only required in case that you want to use interpolated data.

3.1.4 monmean

Calculates the monthly mean for a given variable. See `MonthlyMean`

Warning: This diagnostic does not use the frequency configuration from the config file. You must specify the original frequency when calling it. Otherwise, it will always try to use daily data.

Options:

1. **Variable:** Variable name
2. **Domain:** Variable domain
3. **Original frequency = daily:** Original frequency to use
4. **Grid = “”:** Variable grid. Only required in case that you want to use interpolated data.

3.1.5 relink

Regenerates the links created in the `monthly_mean`, `daily_mean`, etc folders for a given variable. See `Relink`

Options:

1. **Variable:** Variable name
2. **Domain:** Variable domain

3. **Move old = True:** If True, any data founded in the target directory will be moved to another folder (called FOLDER_NAME_old) instead of deleted.
4. **Grid = “:** Variable grid. Only required in case that you want to use interpolated data.

3.1.6 relinkall

Regenerates the links created in the monthly_mean, daily_mean, etc folders for all variables See `RelinkAll`

Options:

This diagnostic has no options

3.1.7 rewrite:

Just rewrites the CMOR output of a given variable. Useful to correct metadata or variable units. See `Rewrite`

Options:

1. **Variable:** Variable name
2. **Domain:** Variable domain
3. **Grid = “:** Variable grid. Only required in case that you want to use interpolated data.

3.1.8 scale

Scales a given variable using a given scale factor and offset ($NEW_VALUE = OLD_VALUE * scale + offset$). Useful to correct errors on the data.

See `Scale`

Options:

1. **Variable:** Variable name
2. **Domain:** Variable domain
3. **Scale value:** Scale factor for the variable
4. **Offset value:** Value to add to the original value after scaling
5. **Grid = “:** Variable grid. Only required in case that you want to use interpolated data.
6. **Min limit = NaN:** If there is any value below this threshold, scale will not be applied
7. **Max limit = NaN:** If there is any value above this threshold, scale will not be applied
8. **Frequencies = [Default_frequency]:** List of frequencies (‘-’ separated) to apply the scale on. Default is the frequency defined globally for all the diagnostics

3.1.9 simdim

Convert *ij* files to lon lat when there is no interpolation required, i.e. lon is constant over *i* and lat is constant over *j*

See `SimplifyDimensions`

Options:

1. **Domain:** Variable domain
2. **Variable:** Variable name
5. **Grid = “:** Variable grid. Only required in case that you want to use interpolated data.

3.1.10 yearlymean

Calculates the daily mean for a given variable. See `YearlyMean`

Warning: This diagnostic does not use the frequency configuration from the config file. You must specify the original frequency when calling it.

Options:

1. **Variable:** Variable name
2. **Domain:** Variable domain
3. **Original frequency:** Original frequency to use
4. **Grid = “:** Variable grid. Only required in case that you want to use interpolated data.

3.2 Ocean

The diagnostics from this section are meant to be used with NEMO variables. Some of them will compute new variables while others just calculate means or sections for variables in the ORCA grid. The interpolation diagnostics are also included here as they are usually used with variables in the ORCA grid.

3.2.1 areamoc

Compute an Atlantic MOC index by averaging the meridional overturning in a latitude band between 1km and 2km or any other index averaging the meridional overturning in a given basin and a given domain. See `AreaMoc`

Warning: The MOC for the given basin must be calculated previously. Usually, it will suffice to call the ‘moc’ diagnostic earlier in the DIAGS list.

Options:

1. **Min latitude:** Minimum latitude to compute
2. **Max latitude:** Maximum latitude to compute
3. **Min depth:** Minimum depth (in levels)
4. **Max depth:** Maximum depth (in levels)
5. **Basin = 'Global':** Basin to calculate the diagnostic on.

3.2.2 averagesection

Compute an average of a given zone. The variable **MUST** be in a regular grid See `AverageSection`

Options:

1. **Variable:** Variable to average
2. **Min longitude:** Minimum longitude to compute
3. **Max longitude:** Maximum longitude to compute
4. **Min latitude:** Minimum latitude to compute
5. **Max latitude:** Maximum latitude to compute
6. **Domain = ocean:** Variable domain

3.2.3 convectionsites

Compute the intensity of convection in the four main convection sites. See `ConvectionSites`

Options:

This diagnostic has no options

3.2.4 cutsection

Cuts a meridional or zonal section. See `CutSection`

Options:

1. **Variable:** Variable to cut the section on
2. **Zonal:** If True, calculates a zonal section. If False, it will be a meridional one
3. **Value:** Reference value for the section
4. **Domain = ocean:** Variable's domain

3.2.5 gyres

Compute the intensity of the subtropical and subpolar gyres. See `Gyres`

Options:

This diagnostic has no options

3.2.6 heatcontent

Compute the total and mean ocean heat content. See `HeatContent`

Options:

1. **Basin** Basin to calculate the heat content one
2. **Mixed layer:** If 1, reduces the computation to the mixed layer. If -1, excludes the mixed layer from the computations. If 0, no effect.
3. **Min depth:** Minimum depth for the calculation in levels. If 0, whole depth is used
4. **Max depth:** Maximum depth for the calculation in levels

3.2.7 heatcontentlayer

Point-wise Ocean Heat Content in a specified ocean thickness. See `HeatContentLayer`

Options:

3. **Min depth:** Minimum depth for the calculation in meters
4. **Max depth:** Maximum depth for the calculation in meters
5. **Basin = 'Global':** Basin to calculate the heat content on.

3.2.8 interpolate

3-dimensional conservative interpolation to the regular atmospheric grid. It can also be used for 2D (i,j) variables. See `Interpolate`

Warning: This interpolation requires the pre-generated weights that can be found in `'/es-nas/autosubmit/con_files/weights'`. Make sure that they are available for your configuration.

Options:

1. **Target grid:** New grid for the data
2. **Variable:** Variable to interpolate
3. **Domain = ocean:** Variable's domain
4. **Invert latitude:** If True, inverts the latitude in the output file.
5. **Original grid = "":** Source grid to choose. By default this is the original data, but sometimes you will want to use another (for example, the 'rotated' one produced by the rotation diagnostic)

3.2.9 interpolateCDO

Bilinear interpolation to a given grid using CDO. See `InterpolateCDO`

Warning: This interpolation is non-conservative, so treat its output with care. It has the advantage that does not require the pre-generated weights so it can be used when the ‘interp’ diagnostic is not available.

Options:

1. **Variable:** variable to interpolate
2. **Target grid:** Variable domain
3. **Domain = ocean:** Variable’s domain
4. **Mask oceans = True:** If True, replaces the values in the ocean by NaN. You must only set it to false if, for some reason, you are interpolating an atmospheric or land variable that is stored in the NEMO grid (yes, this can happen, i.e. with tas).
5. **Original grid = ‘’:** Source grid to choose. By default this is the original data, but sometimes you will want to use another (for example, the ‘rotated’ one produced by the rotation diagnostic)

3.2.10 maskland

Replaces all values excluded by the mask by NaN. See `MaskLand`

Options:

1. **Domain:** Variable to mask domain
2. **Variable:** variable to mask
3. **Cell point = T:** Cell point where variable is stored. Options: T, U, V, W, F
4. **Original grid = ‘’:** Source grid to choose. By default this is the original data, but sometimes you will want to use another (for example, the ‘rotated’ one produced by the rotation diagnostic)

3.2.11 maxmoc

Compute an Atlantic MOC index by finding the maximum of the annual mean meridional overturning in a latitude / depth region. Output from this diagnostic will be always in yearly frequency. See `MaxMoc`

Warning: The MOC for the given basin must be calculated previously. Usually, it will suffice to call the ‘moc’ diagnostic earlier in the DIAGS list.

Warning: This diagnostic can only be computed for full years. It will discard incomplete years and only compute the index in those with the full 12 months available.

Options:

1. **Min latitude:** Minimum latitude to compute
2. **Max latitude:** Maximum latitude to compute
3. **Min depth:** Minimum depth (in levels)
4. **Max depth:** Maximum depth (in levels)
5. **Basin = ‘Global’:** Basin to calculate the diagnostic on.

3.2.12 mixedlayerheatcontent

Compute mixed layer heat content. See `MixedLayerHeatContent`

Options:

This diagnostic has no options

3.2.13 mixedlayersaltcontent

Compute mixed layer salt content. See `MixedLayerSaltContent`

Options:

This diagnostic has no options

3.2.14 moc

Compute the MOC for oceanic basins. Required for ‘areamoc’ and ‘maxmoc’ See `Moc`

Options:

This diagnostic has no options

3.2.15 mxl

Compute the mixed layer depth. See `Mxl`

Options:

This diagnostic has no options

3.2.16 psi

Compute the barotropic stream function. See `Psi`

Options:

This diagnostic has no options

3.2.17 regmean

Computes the mean value of the field (3D, weighted). For 3D fields, a horizontal mean for each level is also given. If a spatial window is specified, the mean value is computed only in this window. See `RegionMean`

Options:

1. **Domain:** Variable domain
2. **Variable:** Variable to average
3. **Grid_point:** NEMO grid point used to store the variable: T, U, V ...
4. **Basin = Global:** Basin to compute
5. **Save 3d = True:** If True, it also stores the average per level
6. **Min depth:** Minimum depth to compute in levels. If -1, average from the surface
7. **Max depth:** Maximum depth to compute in levels. If -1, average to the bottom
8. **Variance = False:** If True, it also stores the variance
9. **Original grid = ‘’:** Source grid to choose. By default this is the original data, but sometimes you will want to use another (for example, the ‘rotated’ one produced by the rotation diagnostic)

3.2.18 rotate

Rotates the given variables See `Rotation`

Options:

1. **Variable u:** Variable’s u component
2. **Variable v:** Variable’s u component
3. **Domain = ocean:** Variable domain:
4. **Executable = /home/Earth/jvegas/pyCharm/cfutools/interpolation/rotateUVorca:** Path to the executable that will compute the rotation

Warning: This default executable has been compiled for ORCA1 experiments. For other resolutions you must use other executables compiled ad-hoc for them

3.2.19 siasiesiv

Compute the sea ice extent , area and volume in both hemispheres or a specified region. See `Siasiesiv`

Options:

1. **Basin = 'Global':** Basin to restrict the computation to.

3.2.20 vgrad

Calculates the gradient between two levels in a 3D ocean variable. See `VerticalGradient`

Options:

1. **Variable:** Variable to compute
2. **Upper level = 1:** Upper level. Will be used as the reference to compute the gradient
3. **Lower level = 2:** Lower level.

3.2.21 verticalmean

Chooses vertical level in ocean, or vertically averages between 2 or more ocean levels. See `VerticalMean`

Options:

1. **Variable:** Variable to average
2. **Min depth = -1:** Minimum level to compute. If -1, average from the surface
3. **Max depth:** Maximum level to compute. If -1, average to the bottom

3.2.22 verticalmeanmeters

Averages vertically any given variable. See `VerticalMeanMeters`

Options:

1. **Variable:** Variable to average
2. **Min depth = -1:** Minimum depth to compute in meters. If -1, average from the surface
3. **Max depth:** Maximum depth to compute in meters. If -1, average to the bottom

3.3 Statistics

3.3.1 climpercent

Calculates the specified climatological percentile of a given variable. See `ClimatologicalPercentile`

Options:

1. **Domain:** Variable's domain
2. **Variable:** Variable to compute diagnostic on
3. **Leadtimes:** Leadtimes to compute
4. **Bins:** Number of bins to use to discretize the variable

3.3.2 monpercent

Calculates the specified monthly percentile of a given variable. See `MonthlyPercentile`

Options:

1. **Domain:** Variable's domain
2. **Variable:** Variable to compute diagnostic on
3. **Percentiles:** List of requested percentiles ('-' separated)

TIPS AND TRICKS

4.1 Working with ORCA1

If you plan to run diagnostics for ORCA1 resolution, be aware that your workstation will be more than capable to run them. At this resolution, memory and CPU consumption is low enough to allow you keep using the machine while running, specially if you reserve a pair of cores for other uses.

4.2 Configuring core usage

By default, the Earth Diagnostics creates a thread for each available core for the execution. If you are using a queueing system, the diagnostics will always use the number of cores that you reserved. If you are running outside a queueing system, the diagnostics will try to use all the cores on the machine. To avoid this, add the `MAX_CORES` parameter to the `DIAGNOSTICS` section inside the `diags.conf` file that you are using.

4.3 NEMO files

Unlike the bash version of the ocean diagnostics, this program keeps the NEMO files in the scratch folder so you can launch different configurations for the same experiment with reduced start time. You will need to remove the experiment's folder in the scratch directory at the end of the experiment to avoid wasting resources. To do this, just use

```
earthdiags -f PATH_TO_CONF --clean
```

If you plan to run the earthdiagnostics only once, you can add this line after the execution

WHAT TO DO IF YOU HAVE AN ERROR

Sometimes, the diagnostics may crash and you will not know why. This section will give you a procedure to follow before reporting the issue. This procedure is intended to solve some common problems or, at least, to help you in creating good issue reports. Remember: a good issue report reduces the time required to solve it!

Hint: Please, read carefully the error message. Most times the error message will point you to the problem's source and sometimes even give you a hint of how to solve it by yourself. And if this is not the case or if you find it obscure, even if it was helpful, please contact the developers so it can be improved in further versions

Try this simple steps BEFORE reporting an issue

- Clean scratch folder
- Update to the latest compatible tag: maybe your issue is already solved in it
- If you get the error for the first chunk of a given diagnostic, change the number of chunks to 1
- Call the diags with the `-lc DEBUG -log log.txt` options

Now, you have two options: if everything is fine, the error was probably due to some corrupted files or some unstable machine state. Nevertheless, try running the diagnostic with `-lc DEBUG -log log.txt` for all the chunks. If everything it's fine that's all.

If you experienced the same problem again, go to the GitLab portal and look into the open issues (https://earth.bsc.es/gitlab/es/ocean_diagnostics/issues). If you find your issue or a very similar one, use it to report your problems. If you can not find an open one that suites your problem, create a new one and explain what is happening to you. In any case, it will be very useful if you can attach your `diags.conf` and `log.txt` files and specify the machine you were using.

After that, it's just a matter of waiting for the developers to do their work and answering the questions that they may have. Please, be patient.

Caution: Of course, there is a third option: you keep experiencing an error that appears randomly on some executions but you are not able to reproduce it in a consistent manner. Report it and attach as much logs and configuration files as you have, along with the date and time of the errors.

DEVELOPER'S GUIDE

The tool provides a set of useful diagnostics, but a lot more can be required at anytime. If you miss something and are able to develop it, you are more than welcome to collaborate. Even if you can not develop it, please let us know what do you want.

The first step is to go to the GitLab page for the project (https://earth.bsc.es/gitlab/es/ocean_diagnostics/) and open a new issue. Be sure that the title is self-explicative and give a detailed description of what you want. Please, be very explicit about what you want to avoid misunderstandings.

Hint: If reading your description, you think that you are taking the developers as stupids, you are doing it perfectly.

Don't forget to add the relevant tags. At this stage you will have to choose between 'enhancement', if you are proposing an improvement on a currently available feature, or 'new feature' in any the other case.

Now, if you are thinking on developing it yourself, please refer to the BSC-ES Git strategy ([wiki_link_when_available](#)) If you have any doubts, or just want help to start the development, contact javier.vegas@bsc.es.

6.1 Developing a diagnostic

For new diagnostics development, we have some advice to give:

- Do not worry about performance at first, just create a version that works. Developers can help you to optimize it later.
- There is nothing wrong with doing some common preparations in the `generate_jobs` of the diagnostic.
- Parallelization is achieved by running multiple diagnostics at a time. You don't need to implement it at diagnostic level
- Use the smallest time frame for your diagnostic: if you can work at chunk level, do not ask for full year data.
- Prefer NCO over CDO, you will have less problems when versions change.
- Ask for help as soon as you get stuck.
- Use always the methods in Utils instead of writing your own code.
- Use meaningful variable names. If you are using short names just to write less, please switch to an editor with autocompletion!
- Do not modify the mesh and mask files, another diagnostic can be using them at the same time.

FREQUENTLY ASKED QUESTIONS

Here will be the answers to the most usual questions. For the moment, there is nothing to see here...

MODULE DOCUMENTATION

8.1 earthdiagnostics

8.1.1 earthdiagnostics.box

8.1.2 earthdiagnostics.cdftools

8.1.3 earthdiagnostics.cmorizer

8.1.4 earthdiagnostics.cmormanager

8.1.5 earthdiagnostics.config

8.1.6 earthdiagnostics.constants

8.1.7 earthdiagnostics.datamanager

8.1.8 earthdiagnostics.diagnostic

8.1.9 earthdiagnostics.earthdiags

8.1.10 earthdiagnostics.parser

8.1.11 earthdiagnostics.utils

8.1.12 earthdiagnostics.variable

8.2 earthdiagnostics.general

8.2.1 earthdiagnostics.general.attribute

8.2.2 earthdiagnostics.general.dailymeans

8.2.3 earthdiagnostics.general.module

8.2.4 earthdiagnostics.general.monthlymean

8.2.5 earthdiagnostics.general.relink

8.2.6 earthdiagnostics.general.relinkall

8.2.7 earthdiagnostics.general.rewrite