

CMOR2 SPECS patch documentation

Pierre-Antoine Bretonnière (pierre-antoine.bretonniere@ic3.cat), IC3, June 2013

I-Introduction

II-Updates from v.2.8.3 to SPECS patch

1)New arguments in the functions calls

2)Updated functions

3)Updated files

4)Update MIP Tables

III-Use of double time axis

IV-Backward compatibility

I-Introduction:

The SPECS project (Seasonal-to-decadal climate Prediction for the improvement of European Climate Services, <http://www.specs-fp7.eu/SPECS/Home.html>) will deliver a new generation of climate prediction systems for seasonal-to-decadal time scales, to provide actionable climate information for a wide range of users.

Across the different experiments run during this project, a huge amount of files is to be produced (the lower estimate of the volume of data to be stored is close to 80Tb), stored at a common repository and shared among partners. To allow any exchange and comparison of data, all the files need to be written in a common format and stored with a unified repository tree. A document (http://www.specs-fp7.eu/wiki/index.php/File:SPECS_standard_output.pdf) has been written to define all these conventions (file format, attributes, variable list, structure of the repository tree...).

To be sure that all the files that are going to be stored have the right format, the Climate Model Output Rewriter (CMOR) library can be used to post-process the output of the ocean and atmospheric models, in a same way than it was for CMIP5 project.

But, as the conventions from SPECS differ from the CMIP5 ones, there was a need to produce an update from CMOR2 v2.8.3 library used for CMIP5 to comply these changes. The main change is the introduction of a double time axis coordinate (reference time and lead time) in the output files. Some other changes in the global attributes of the files as well as the directory tree under which the data has to be stored made mandatory to develop a patch to the library.

The following fully documents the difference between the two versions, the new arguments that some functions need, and an explanation of how to use the new functions in the CMORization programs to be able to produce the double time axis. To avoid the development of parallel specific project versions, this new patch is fully compatible with the CMIP5 programs and the previous CMIP5 CMORization programs can still be transparently used with this version of the library.

II-Updates from v.2.8.3 to SPECS patch

1) New arguments in function call:

This is the most important part of the changes that the user needs to know about. Indeed, when used for SPECS CMORization, some functions will require some different arguments than with the previous version. As before, a check will be done when the functions are called and if the wrong arguments are passed (missing arguments, wrong type, or SPECS arguments passed while in CMIP5 mode for example), some warnings and/or error messages will be prompted. In the next table and all along the document thereafter, the functions name are the C/Fortran name, keep in mind that for the Python interface, they behave the same way except they do not have the “cmor_” prefix. In all the document, even if only the C/Fortran name of the functions is mentioned, everything stays valid for the three languages.

Function arguments	V2.3.8 (CMIP5)	V3.0.1 (SPECS)
cmor_setup	inpath, netcdf_file_action, set_verbosity, exit_control, logfile, create_subdirectories	inpath, netcdf_file_action, set_verbosity, exit_control, logfile, create_subdirectories , [specs=1]
Comment:	This optional specs keyword is used to switch between CMIP5 and SPECS modes (see section IV-Backward compatibility below)	
cmor_dataset	outpath, experiment_id, institution, source, calendar, [realization=1], [contact], [history], [comment], [references], [leap_year], [leap_month], [month_lengths], [model_id], [forcing], [initialization_method], [physics_version], [institute_id], [parent_experiment_id], [branch_time], [parent_experiment_rip]	outpath, experiment_id, institution, source, calendar, [realization=1], [contact], [history], [comment], [references], [leap_year], [leap_month], [month_lengths], [model_id], [forcing], [initialization_method], [physics_version], [institute_id], [parent_experiment_id], [branch_time], project_id, initialization_description, forecast_ref_time, associated_model, physics_description, series, [parent_experiment_rip]
Comment:	As the global attributes and the structure of the repository tree differ between the two versions, new arguments have to be passed to define the dataset. These arguments will be read, as the ones previously present in the function, in the namelist passed to the program using the CMOR library.	
cmor_write	var_id, data, [file_suffix], [ntimes_passed], [time_vals],	var_id, data, [file_suffix], [ntimes_passed], [time_vals],

	[time_bnds], [store_with]	[time_bnds], reftime_vals , [store_with]
Comment:	As the files now contain two time axes (see section IV-Use of double time axis below), the cmor_write function needs to be passed the new reftime values, computed externally.	

2) Updated functions:

To help the user to make the transition and understand the changes between v2.8.3 (or any CMIP5 related CMOR library version) and the SPECS-CMOR patch, this is the complete list of updated functions.

- cmor_setup: adding the keyword argument SPECS to ensure the use of backward compatibility.
- cmor_set_cur_dataset_attribute: update the global attributes list. Project_id, initialization_description, forecast_ref_time, associated_model, physics_description, and series have been added to the list of mandatory attributes for SPECS mode while institute_id has been removed.
- cmor_dataset: same thing as cmor_set_cur_dataset_attribute.
- cmor_write: add reftime as argument of the function (1D-array containing the values of the reftime, computed before by an external function), define and write the reftime variable for each variable.
- cmor_create_output_path: modify the repository tree structure adding the new levels.
- cmor_close: remove false error message for time axis.
- cmor_variable: change time variable name.
- cmor_write_var_to_file: remove obsolete test about monotonicity for leadtime, change units name from “days from ...” to “days”.
- cmor_func_def.h: update list of arguments in the modified functions.
- cmor.h: change CMOR version numbers, add definition of the CMOR_SPECS parameter, update list of arguments in the modified functions.

3) Updated files:

All the functions above mentioned belong to the following files, this is the list of modified files.

- cmor.c: contains the modifications of all above mentioned functions except cmor_variable and cmor_write_var_to_file
- cmor_cfortran_interface.c: duplicate the functions whose list of arguments have been modified between the two versions (cmor_setup, cmor_dataset, cmor_write): the fortran interface now calls the C functions through different functions according if we are in SPECS or CMIP5 mode.
- cmor_fortran_interface.f90: same thing as cmor_cfortran.c
- cmormodule.c: declare new variables for functions with new arguments.
- cmor_variable.c: contains the modifications of cmor_variable and cmor_write_var_to_file functions.

4) Other updates:

a) MIP Tables:

In addition to the changes in the CMOR2 library itself, the update to CMOR-SPECS requires new MIP tables: SPECS_Amon, SPECS_OImon, SPECS_Omon, SPECS_day. These new tables names have to be passed when calling the `cmor_load_table` and `cmor_set_table` functions. Compared with the previous CMIP5 or CORDEX tables, these are the main changes brought to the tables:

- Table header: all the MIP tables have, before the axis and variables parameters, a header with information such as the `cmor_version` that they are written for, required attributes and experiments name acceptable. Due to the new requirements of the CMOR-SPECS functions, have been updated the following entries:
 - `cmor_version`
 - `cf_version`: from 1.4 to 1.6 to comply with the latest Climate and Forecast (CF) conventions.
 - `table_date`
 - `required_global_attributes` (see `cmor_dataset` new argument list)
 - `expt_id_ok`: the new names of the experiments
- Update list of variables: the complete list of variables asked for the various experiments of SPECS can be found on the SPECS wiki page (http://www.specs-fp7.eu/wiki/index.php/Data#SPECS_list_of_variables). All these variables with all their details have been updated in the according tables.
- Add double time axis reference
- Add NC4 compression parameters: in the previous comparison model intercomparison exercises, the output format required was NetCDF3, hence with no compression option allowed for the variables. As in SPECS, the output required format is now NetCDF4, it allows some compression through the `netcdf nc_def_var_deflate` function (http://www.unidata.ucar.edu/software/netcdf/docs/netcdf-c/nc_005fdef_005fvar_005fdeflate.html). To use this function, three arguments are needed: `shuffle`, `deflate`, `deflate_level`. These are the parameters that are defined in the new tables. For every single experiment, the user can still tune more finely these parameters to get more efficient compression ratios.

b) External program Namelist:

Compared to the previous version, some new arguments have been introduced in some functions, especially in the `cmor_dataset` (`project_id`, `initialization_description`, `forecast_ref_time`, `associated_model`, `physics_description`, `series`). Usually, these new arguments have to be read by the program using the CMOR2 library in an external Namelist.

In this Namelist, are also specified other arguments such as the experiment name. The experiment names for the SPECS project differ from the previous projects. To be sure that the good names are being specified in the Namelist and then in `cmor_dataset` arguments, all the standard names of the experiments can be found in the SPECS conventions documentation (http://www.specs-fp7.eu/wiki/index.php/File:SPECS_standard_output.pdf). Anyway, all the experiment names have been added to the MIP tables so that a compliance check can be performed when the outpath is created.

IV-Use of double time axis:

One of the major novelty of the SPECS conventions is the add of a double time axis: instead of containing a single time axis called time(time), the SPECS CMORized files are supposed to have two variables for time: reftime(time) and leadtime(time). Note that the dimension is preserved and still called time, only the axis/variable have changed.

The reftime variable (long_name = "forecast reference time", standard_name = "forecast_reference_time", units= "days from 1850-01-01") indicates at what date the experiment started (and not from which date the data of the file does). It refers to the start_date of the experiment.

The leadtime variable (long_name = "Time elapsed since the start of the forecast", standard_name = "forecast_period", units = "days") is the equivalent of the time variable used previously in the CMIP5 conventions.

These two variables are needed because in a forecasting context, a single verification time (say the 1 December 2012) might have different forecasts (e.g. the 62nd day of a forecast started on the 1st of October 2012 and the 31st day of a forecast started on the 1st of November). If the start of the simulations is only available in the global attributes, a user merging in the same file the two forecasts will miss this essential information.

As, of course, the models won't be modified to add this new axis to the output files, the CMOR-SPECS patch will take care of this. The cmor_write functions has been modified in consequence to manage this new feature. The arguments time_vals and time_bnds remain in the call and now refer to the leadtime. All the attributes (standard, long-name, units...) will be checked to comply the new conventions. To add the reftime, the users now have to specify a new argument when they call the function: reftime_vals=array(size(time_vals)). It corresponds to the values of the reference time. This array can be computed by the program calling the library in a way very similar to how the time/leadtime was computed before (except that the values will be the number of days between 1850-01-01 and the start_date of the experiment). Note that, in a single experiment (same start date), all the values of the reftime will be equal.

VI-Backward compatibility:

Due to the great number of partners that are still using the CMOR2 library in programs to CMORize data with the CMIP5 conventions, developing a new different version of the library which would be project-dedicated (SPECS in this case) would create some new parallel branches of the code that in the end would become very difficult to maintain and to add support to.

In this perspective, this patch has been designed to be completely backward compatible with v2.8.3 used for CMIP5. That is to say that the users using v2.8.3 in CMIP5 CMORization programs, cwill still be able to use this new version without any changes. The arguments of all the functions stay the same, the variable attributes and the structure of the directory will remain unchanged from what they were in the previous version.

To take advantage of the new features of this patch and use it in SPECS CMORization programs, the user will just have to set a new parameter (specs) equal to 1 in the call to cmor_setup function:

```
cmor_setup(inpath=inpath,netcdf_file_action=CMOR_APPEND,
set_verbosity=CMOR_NORMAL, exit_control=CMOR_NORMAL, logfile, create_subdirectories
specs=1). Then, all the other cmor functions called afterwards will request the new arguments and will produce the new directory structure, variable attributes, double time axis etc...
```

In both cases (specs=1 or not), all the functions will have an argument checker to be sure that they

are called in the good mode, i.e. with the good arguments.

Another way to keep the CMIP5 version of the library is to call `cmor_setup` with the argument `specs` set to 0.