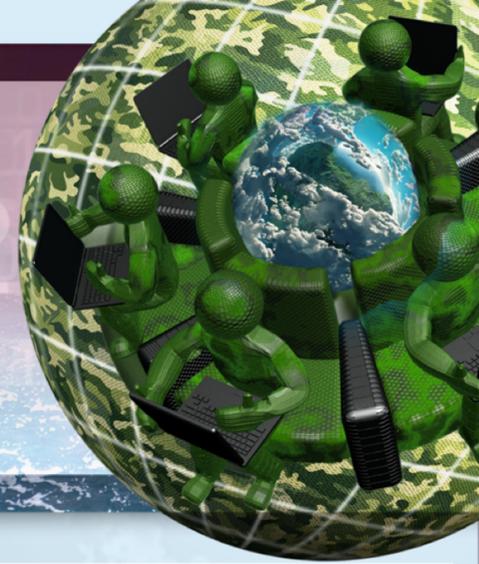


CMOR: Climate Model Output Rewriter

Chris Mauzey, Charles Doutriaux, Karl E. Taylor, Denis Nadeau, Paul J. Durack

Lawrence Livermore National Laboratory



Introduction

The Climate Model Output Rewriter (CMOR) is a library used for generating CF-compliant NetCDF files to facilitate model output intercomparison for various Model Intercomparison Projects (MIPs).

It is written in C with bindings to Fortran and Python, and is currently supported on Linux and macOS. CMOR also includes PrePARE, a Python-based program used for validating whether NetCDF files conform to the Coupled Model Intercomparison Project Phase 6 (CMIP6) standards for publication into the Earth System Grid Federation (ESGF) platform.

Features of CMOR3

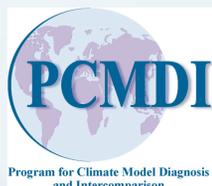
The purpose of CMOR is to give researchers a way to convert climate model data to a common format within the standards of a Model Intercomparison Project (MIP). The output of CMOR has the following characteristics:

- Each file written is for a single variable of a single model for a single experiment along with accompanying coordinates, attributes, and metadata.
- Time slices of an experiment can be stored in a single file, or in a sequence of files.
- Most of the metadata in the output files comes from the MIP tables, which are in JSON format. These tables define the attributes and requirements of variables.

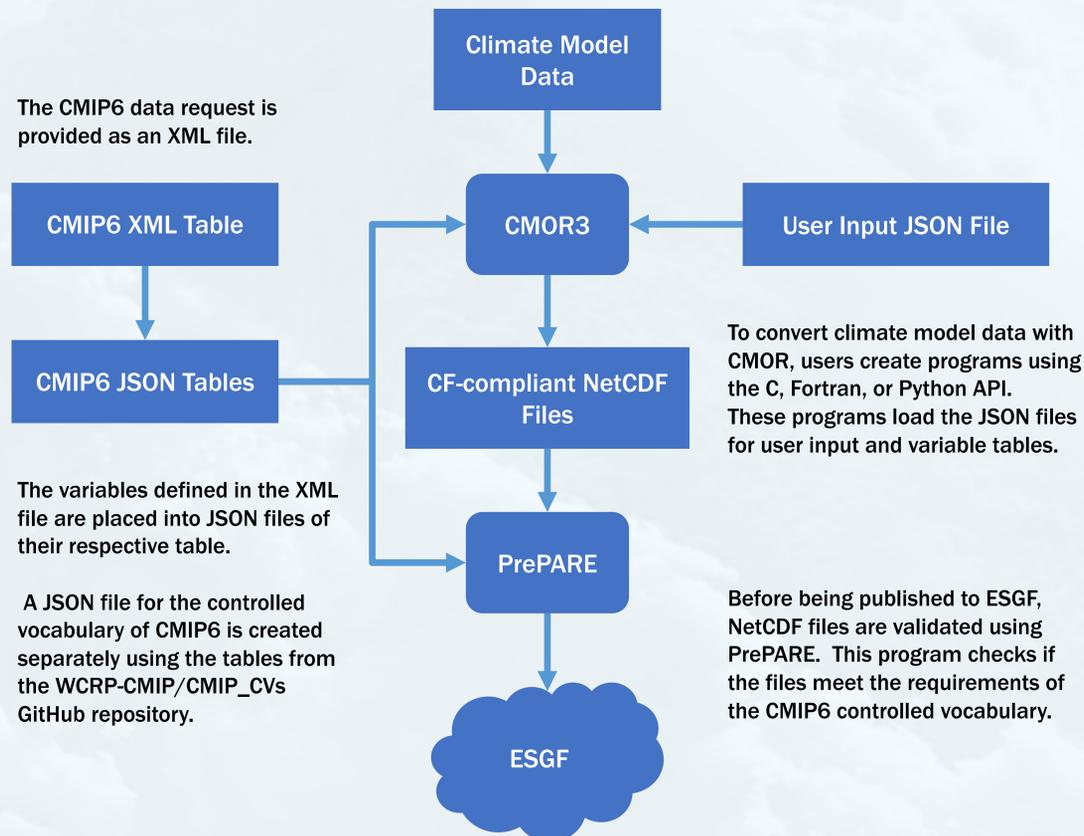
CMOR stores output in the NetCDF format. By default, it uses the NetCDF4 library to take advantage of compression and chunking. However, CMOR can be setup to write files that can be read with NetCDF3. [1]

Converting model data with CMOR3 involves creating a program with either the C, Fortran, or Python API. Programs typically have the following structure:

- Set up CMOR with options to control error handling and logging
- Set user input with a JSON file that defines parameters such as output directory, data source, experiment ID, etc.
- Load a MIP table for the variables being written
- Define coordinate and time axes
- Define variables
- Write variables and associated data to NetCDF files



ESGF Publishing Workflow



Software Development and Distribution

CMOR3 and PrePARE are both open source projects that are available in a repository on GitHub. The repository is maintained by the PCMDI team at LLNL but is also open to outside researchers for posting issues and pull requests. [2]

The CMIP6 data request is maintained by Dr. Martin Juckes at the Science and Technology Facilities Council (STFC) in Swindon, UK. [3] The data request XML file is processed by scripts found in the xml-cmor3-database GitHub repository [4] to create CMIP6 JSON tables, which are saved to the cmip6-cmor-tables repository [5].

The CMIP6 controlled vocabulary file (CMIP6_CV.json) is built from the JSON files from the World Climate Research Programme's CMIP_CVs GitHub repository. [6] The controlled vocabulary file is kept updated by a daily automated process that checks for changes in CMIP_CVs. Aside from CMIP6, CMOR3 is also used by the obs4MIPs [7] and input4MIPs [8] projects.

The stable and nightly releases of CMOR3 are available for Anaconda Python via Conda. Nightly releases are created after changes are pushed to the master branch of CMOR3. When changes are pushed to the master branch, continuous integration tests are run on CircleCI and Travis CI. Upon successful completion of tests, a new nightly release is pushed to the Anaconda Cloud repository for both Linux and macOS. CMOR3 currently only supports Python 2.7. [9]



PrePARE

PrePARE, which stands for "Pre-Publication Attribute Reviewer for ESGF", is a Python-based command line program that is installed alongside CMOR3 that is used for validation of NetCDF files to CMIP6 standards. It validates files by comparing their attributes with those listed in CMIP6_CV.json, a file that contains the controlled vocabulary for CMIP6.

Validation will only allow files with institution, source, and experiment IDs that have been registered for CMIP6 publication into ESGF. It will also make sure that attributes such as grid resolution, creation time, and the "further info" URL are within standards.

PrePARE can handle input of either a single file, or a directory that can be searched recursively for files. It can process multiple files in parallel through the use of Python's multiprocessing module.

Future Development

Features being considered for future versions of CMOR:

- Streamlining Python unit tests
- Generalization of PrePARE to handle obs4MIPs and input4MIPs data
- Support for Python 3

Acknowledgements

We have benefited from contributions by Emma Hogan, Guillaume Levavasseur, Jean-Yves Peterschmitt, Fabian Wachsmann, and Stéphane Sénési, among others, who have discovered bugs, submitted code to fix bugs or augmented capabilities, and made valuable suggestions for improving CMOR and PrePARE.

References

1. CMOR3 Documentation, <https://cmor.llnl.gov/>
2. CMOR GitHub, <https://github.com/PCMDI/cmor>
3. The WGCN Infrastructure Panel, <https://earthsystemcog.org/projects/wip/aboutus/>
4. xml-cmor3-database, <https://github.com/PCMDI/xml-cmor3-database/>
5. cmip6-cmor-tables, <https://github.com/PCMDI/cmip6-cmor-tables/>
6. CMIP_CVs, https://github.com/WCRP-CMIP/CMIP6_CVs
7. obs4MIPs, <https://github.com/PCMDI/obs4MIPs-cmor-tables>
8. input4MIPs, <https://github.com/PCMDI/input4MIPs-cmor-tables>
9. CMOR Anaconda, <https://anaconda.org/pcmdi/cmor>