

Tools developed by IPSL and IPSL partners

Many software programs are developed at IPSL or at IPSL's partners (see list below).

Table of Content

Tools developed by IPSL and IPSL partners	1
1. modipsl	2
2. libIGCM	2
3. XIOS	2
4. IOIPSL and REBUILD	2
4.1. IOIPSL	2
4.2. Rebuild	2
5. Tools to prepare a coupled configuration	2
5.1. MOZAIC	2
5.2. MOSAIX	2
5.3. CPLRESTART	3
5.4. Bug bilinear	3
6. AddNoise	3
7. FAST/ATLAS	3
8. WEB applications	3
8.1. Trusting	3
8.2. Intermonitoring	3
9. OASIS	3
10. External Tools	4

1. modipsl

modipsl is a suite of scripts which allow you to :

- obtain a given model configuration from a server;
- compile a configuration on a specific machine.

More information on how to retrieve and use modipsl [can be found here](#).

2. libIGCM

libIGCM is a running environment library which allow you to run a simulation on a specific machine by launching computing and post-processing jobs. Detailed information about **libIGCM** can be found [here](#).

3. XIOS

XIOS (XIOS stands for XML-IO-SERVER) is an input/output library which allows you to manage parallel I/Os in climate models in a flexible and efficient way. XIOS (version ≥ 2.5) is currently used in all of IPSL components. Here are XIOS main features :

- library dedicated to I/O management of climate code, developed at IPSL.
- XML configuration file.
- attached mode (library) or server mode (asynchronous transfer), multiple (sequential writing) or single (parallel writing) output file.
- NetCDF format.

More information can be found [here](#).

4. IOIPSL and REBUILD

4.1. IOIPSL

IOIPSL is an I/O library for the IPSL models. This library is not being further developed (except to fix bugs). The XIOS library has replaced IOIPSL in the IPSL models for output handling. IOIPSL is still used in IPSL components for restart and parameters input actions.

4.2. Rebuild

Rebuild is a tool which allows you to concatenate model output files in NetCDF format. At the end of a simulation, files for each process are automatically (thanks to libIGCM) assembled to a global grid into one single output file. Rebuild is developed at IPSL and is a fortran code included in IOIPSL extended version.

For each machine a Rebuild executable is provided in the shared tools folder.

Thanks to XIOS output library (parallel writing functionality) rebuild is not used anymore in v6 configuration.

5. Tools to prepare a coupled configuration

Tools to creates interpolations between ocean and atmospheres grid, and initial state files for the coupler.

5.1. MOZAIC

MOZAIC is the legacy code used to build interpolation weights between ocean and atmosphere. It should be considerer as obsolete. It's a messy sequential code. Poor documentation available [here](#).

5.2. MOSAIX

MOSAIX is the new code that creates the interpolation weights between ocean and atmosphere. It's a set a bash, python and fortran code that relies on XIOS for computing the weigths. It can be used in parallel. See [Mosaix](#).

5.3. CPLRESTART

CPLRESTART is a small bash that builds (using [nco utilities](#)) the restart file for OASIS. It needs outputs from forced configurations (grid_T file from NEMO, histmth file from LMDZ).

5.4. Bug bilinear

In old coupled configurations, the wind stress is interpolated with the bilinear interpolation scheme of the [Scrip](#) library included in OASIS-MCT.

In IPSLCM6 using ORCA1, the bilinear interpolation of the wind stress is bugged. A small set of python routines has been developed to overcome the problem : see <http://forge.ipsl.jussieu.fr/igcmg/wiki/IPSLCM6/Coupling/BugBilinear>.

New coupled configurations are not affected, as they use weights computed by MOSAIX (see above).

6. AddNoise

AddNoise allows you to apply an equi-distributed white noise which amplitude can be specified in a restart file in order to prepare ensemble experiments. See also : [Doc/Setup/Ensemble](#)

7. FAST/ATLAS

FAST is a library to facilitate post-processing and visualization with Ferret.

FAST is a collection of Ferret scripts and K-shell scripts. Its purpose is to hide a certain number of low-level and repetitive commands which can transform your Ferret scripts into long and unclear programs. With the scripts provided here, you will more easily control and set pages, viewports, texts, legends, colorbars, logos, ratios, zooms and also produce polygon maps and vectors maps.

Ferret is a very powerful application developed by the PMEL. We strongly recommend first reading the documentation and the tutorial they provide from [their web site](#).

ATLAS repository is a collection of useful files (scripts, palette files, configuration files, etc.) to help produce plots to visualize and analyze output from the different components of the Earth System Model (ESM) from Institut Pierre Simon Laplace (IPSL), France. ATLAS is mainly base on FAST scripts.

Detailed information about Fast and Atlas can be found at [fast](#) and [atlas](#).

8. WEB applications

web site to access to IPSL's Trusting and Intermonitoring software <http://webservices.ipsl.fr>

8.1. Trusting

The [trusting](#) is a software use by IPSL model's developpers to check the reliability of their codes.

8.2. Intermonitoring

The https://github.com/PBrockmann/interMonitoring_intermonitoring is closely related to the monitoring. Read more in chapter *Simulation and post-processing*: [Doc/Running](#).

See also information at webservices page : <http://webservices.ipsl.fr/monitoring>

9. OASIS

OASIS is a tool developed at Cerfacs (Toulouse, France) which allows you to couple different climate model components. It allows the synchronized exchange and the interpolation of information between model components. OASIS is used in the IPSL Earth System model to couple NEMO (the ocean model) and LMDZ (the atmospheric model). The modipsl and libIGCM tools manage the OASIS setup and compilation. OASIS3 (sequential coupler) was used in IPSLCM5 coupled model. OASIS3-MCT (parallel coupler) is currently used in IPSLCM6 coupled model.

More information on OASIS is available [here](#).

10. External Tools

[Important tools used in the IPSL climate modeling framework](#)