

## Main commands for using a model configuration via modipsl-libIGCM

### WARNING

All commands are given for the IPSLCM6 configuration (LR resolution - historical experiment) for an experiment called MyJobTest. You need to adapt them to your own case. If you want more details on some parts, see the full documentation on the IPSL wiki <https://forge.ipsl.jussieu.fr/igcmg/doc/wiki/Doc>

### Extract modipsl

Connect to a computer (TGCC - IDRIS - Ciclad - Obelix)

From a terminal, use svn to extract modipsl:

```
svn co http://forge.ipsl.jussieu.fr/igcmg/svn/modipsl/trunk modipsl
```

### How to list the model configurations managed by modipsl

Use the `-h` option of the `model` command in the `util/` directory. This command will browse the `mod.def` file and show all available model configurations.

```
cd modipsl/util
./model -h
```

### Extract a model configuration

Once you have chosen your model configuration, use the `model` command in the `util/` directory to extract it. For example for the IPSLCM6.2.2 configuration:

```
cd modipsl/util
./model IPSLCM6.2.2
```

## Compilation

Compilation is done with a script named `compile_****.sh` and stored in `modipsl/config/****/`. Use the `-h` option to show all the options to compile your configuration. Then launch the script `compile_****.sh` to compile your configuration with the chosen options (or none if you want the default).

```
cd modipsl/config/IPSLCM6
./compile_ipslcm6 -h
./compile_ipslcm6.sh [resolution_option] [debug_option] [other_option]
```

Executable files will be found in `modipsl/bin/` directory.

## Create the experiment directory

The `modipsl/config/****/EXPERIMENTS/` directory contains default experiment directories. Choose one of them and copy its `config.card` file to create your own experiment directory:

```
cd modipsl/config/IPSLCM6
cp EXPERIMENTS/IPSLCM/historical/config.card .
```

Change the `JobName` in `config.card`. The `modipsl/libIGCM/ins_job` script will create the experiment directory with its own job file. Answer all the questions to complete the job header.

```
vi config.card    # modify JobName to JobName=MyJobTest
../../libIGCM/ins_job
```

An experiment directory (`MyJobTest/`) is created. It contains 3 files: `config.card`, `Job_MyJobTest` and `run.card.init` and 4 directories: `COMP/`, `PARAM/`, `DRIVER/` and `POST/`.

## Prepare the experiment

In the `config.card` file, modify these sections:

SpaceName	Type of experiment (DEVT, TEST or PROD)
DateBegin	Start date of simulation (first day) format: year-month-day. Ex: 1950-01-01
DateEnd	End date of simulation (last day included in the simulation) Ex:1959-12-31
PeriodLength	Length of each simulation period (1D, 5D, 1M, 3M, 1Y, 10Y,...)

The `COMP/` directory includes a card file per component (model) of the configuration. Each card file contains several categories to organize input and output files for this model configuration.

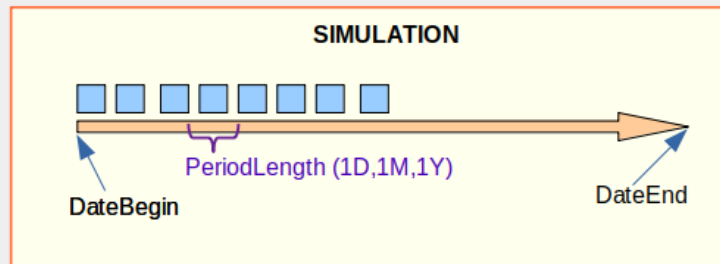
[InitialStateFiles]	list of files used to set initial state (ex: start.nc and startphy.nc for lmdz)
[BoundaryFiles]	list of files describing boundary conditions (list and listNonDel)
[ParametersFiles]	List of input text files (*.def, *.xml, namelist, etc)
[RestartFiles]	list of files required for restart operation of this component (never modify this part)
[OutputText]	list of output text files (stored in Debug directory if the simulation failed)
[OutputFiles]	List of output files to be stored on the file server.
[Post_...]	description of post-processing

The `PARAM/` directory includes the parameter files needed for each component (model) of the configuration. The `DRIVER/` directory includes the driver files needed to prepare each step of the experiment before and after the launch of the executable.

## The experiment job

The job is created at the same time as the experiment directory by the `ins_job` command. If needed, modify the `PeriodNb` parameter to fit with your simulation.

PeriodNb = number of simulation periods that can be executed over the cpu time



## How to submit the job

Use one of these commands (depending on your machine) to submit the job `Job_MyJobTest` in `modips1/config/IPSLCM6/MyJobTest/`:

```
JeanZay (IDRIS) : sbatch Job_MyJobTest
Irene (TGCC)    : ccc_msub Job_MyJobTest
Obelix, Ciclad : qsub Job_MyJobTest
```

## How to track the status of your simulation

- To check the job status (if your simulation is pending/running/completed):

```
JeanZay (IDRIS) : squeue -u $USER
Irene (TGCC)    : ccc_mpp -u $USER
Obelix, Ciclad  : qstat -u $USER
```

- History of the status of your simulation is kept in the `run.card` file in `MyJobTest/`.

## Output directories

Depend on the `SpaceName` (TEST, DEVT or PROD) in `config.card`:

SpaceName	Computing Center	File Server
TEST	TGCC	<code>\$CCCSCRATCHDIR</code>
TEST	IDRIS	<code>\$SCRATCH</code>
DEVT / PROD	TGCC	<code>\$CCCSTOREDIR</code>
DEVT / PROD	IDRIS	<code>\$STORE</code>

The output files will be located in the components subdirectories (e.g. `ATM/Output`, `ATM/Analyse`) in the following standardized tree structure:

`IGCM_OUT/TagName/[SpaceName]/[ExperimentName]/JobName/`

## How to continue a simulation

First, change the `DateEnd` in `config.card` (do not change `DateBegin`).

Then change `PeriodState=Completed` to `PeriodState=OnQueue` in `run.card`.

Submit the job.

## How to clean or purge your simulation

- Clean up everything related to the **last period that failed**:

```
../../../../libIGCM/clean_PeriodLength.job #Read questions and answer yes to  
erase the files.
```

`run.card` must exist and `PeriodState` must be equal to `Fatal`.

- Clean up the **whole simulation** (warning: whole simulation outputs will be permanently erased):

```
../../../../libIGCM/purge_simulation.job #Read questions and answer yes to  
erase the files.
```

`run.card` must exist, regardless of what is in `PeriodState`.