

## Running simulation and post-processing

---

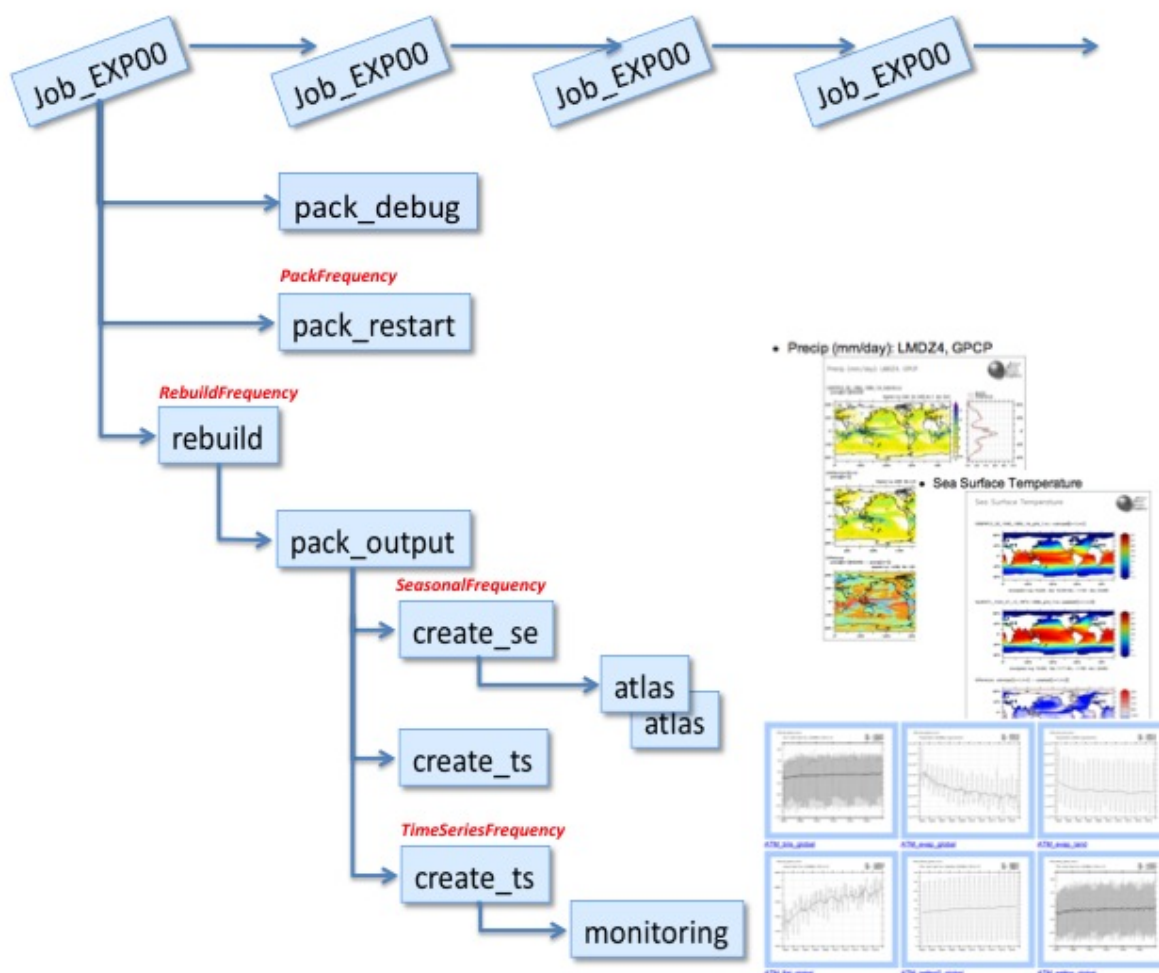
In this chapter you will learn about how to start a simulation and how to use the IPSL models and tools, from the beginning of the simulation to the post processing and basic visualization of the outputs.

### Table of Content

<b>Running simulation and post-processing</b>	<b>1</b>
<b>1. Overview of IPSL running environment workflow</b>	<b>2</b>
<b>2. Simulation - Computing part</b>	<b>2</b>
2.1. Submitting your simulation	2
2.1.1. Launch a simulation test	3
2.2. Status of the running simulation	3
2.2.1. run.card during the simulation	3
2.2.2. Execution directory	3
2.2.3. Accounting mail	3
2.3. End of the simulation	4
2.3.1. messages received	4
Example of message for a successfully completed simulation	4
Example of message when the simulation failed	4
2.3.2. run.card at the end of a simulation	4
2.3.3. Script_Output_JobName	5
2.4. The output files	5
2.4.1. Here is the storage directory structure of the output files produced at TGCC	6
2.4.2. Here is the storage directory structure of the output files produced at IDRIS	6
2.5. Debug/ directory	7
2.6. How to continue or restart a simulation?	7
<b>3. Simulation - Post processing</b>	<b>8</b>
3.1. Post processing in config.card	8
3.2. Rebuild	8
3.3. Concatenation of "PACK" outputs	8
3.3.1. How are the different kinds of output files treated ?	9
3.4. Time Series	9
3.5. Monitoring and intermonitoring	9
3.5.1. Adding a variable to the monitoring	11
3.5.2. Inter Monitoring	12
3.5.3. Mini how to use the intermonitoring	14
3.5.4. How to save the intermonitoring permanently	14
3.6. Seasonal means	15
3.7. Atlas	15
3.8. Storing files like ATLAS, MONITORING and ANALYSE	16
3.9. How to check that the post processing was successful	16

## 1. Overview of IPSL running environment workflow

The main computing job automatically runs post processing jobs (at different frequencies) during the simulation. Here is a diagram describing the job sequence:



## 2. Simulation - Computing part

### 2.1. Submitting your simulation

Once you have [defined and setup your simulation](#) you can submit it. The run commands are:

- `ccc_msub` at TGCC
- `sbatch` at IDRIS

```

irene > ccc_msub Job_MYJOBNAME
jean-zay > sbatch Job_MYJOBNAME

```

These commands return a job number that can be used with the machine specificities to manage your job. Please refer to the [Computing Center](#) page of your machine.

Before starting a simulation it is very important to double check that it was properly setup. We strongly encourage you to perform a short [test](#) before starting a long simulation.

The job you just submitted is the first element of a sequence of jobs. These jobs include the computing job itself, post processing jobs like: *pack*, *create\_ts*, *create\_se* and visualization jobs like *monitoring* and *atlas* which are started at given frequencies.

If you recompile the modele during a simulation, the new executable will be used in the next period of the running job.

### 2.1.1. Launch a simulation test

On config.card :

```
PeriodLength=1D ou 5D
SpaceName=TEST
TimeSeriesFrequency=NONE
SeasonalFrequency=NONE
```

Run on test queue:

modify beginning of main job :

- irene :

```
#MSUB -Q test
#MSUB -T 1800
```

- jean-zay :

```
#SBATCH --time=00:30:00          # Wall clock limit (seconds)
#SBATCH --qos=qos_cpu-dev
```

---

## 2.2. Status of the running simulation

### 2.2.1. run.card during the simulation

A *run.card* file is created as soon as your simulation starts. It contains information about your simulation, in particular the *PeriodState* parameter which is:

- *Start* or *OnQueue* if your simulation is queued
- *Running* if your simulation is being executed
- *Completed* if your simulation was successfully completed
- *Fatal* if your simulation was aborted due to a fatal error

### 2.2.2. Execution directory

At TGCC your simulation is performed in a `$CCCSCRATCHDIR/RUN_DIR/job_number` directory. At IDRIS it's on a `$SCRATCH/RUN_DIR/job_number` directory. You can check the status of your simulation in this directory.

### 2.2.3. Accounting mail

You receive a mail « Simulation Accounting » that indicates the simulation starts fine, how many *PeriodNb* you can use to be efficient and how many computing hours the simulation will consume. For example :

```
Dear Jessica,

this mail will be sent once for the simulation CURFEV13 you recently submitted

The whole simulation will consume around 10074.036500 hours. To be compared with your project allocation.

The recommended PeriodNb for a 24 hours job seems to be around 38.117600. To be compare with the current setting (Job_CURF

Greetings!
```

## 2.3. End of the simulation

### 2.3.1. messages received

#### Example of message for a successfully completed simulation

```
From : no-reply.tgcc@cea.fr
Object : CURFEV13 completed

Dear Jessica,

Simulation CURFEV13 is completed on supercomputer irene3820.
Job started : 20000101
Job ended   : 20001231
Output files are available in /ccc/store/.../IGCM_OUT/IPSLCM5A/DEVT/pdControl/CURFEV13
Files to be rebuild are temporarily available in /ccc/scratch/.../IGCM_OUT/IPSLCM5A/DEVT/pdControl/CURFEV13/REBUILD
Pre-packed files are temporarily available in /ccc/scratch/.../IGCM_OUT/IPSLCM5A/DEVT/pdControl/CURFEV13
Script files, Script Outputs and Debug files (if necessary) are available in /ccc/work/.../modips1/config/IPSLCM5_v5/CURFEV13
```

#### Example of message when the simulation failed

```
From : no-reply.tgcc@cea.fr
Object : CURFEV13 failed

Dear Jessica,

Simulation CURFEV13 is failed on supercomputer irene3424.
Job started : 20000101
Job ended   : 20001231
Output files are available in /ccc/store/.../IGCM_OUT/IPSLCM5A/DEVT/pdControl/CURFEV13
Files to be rebuild are temporarily available in /ccc/scratch/.../IGCM_OUT/IPSLCM5A/DEVT/pdControl/CURFEV13/REBUILD
Pre-packed files are temporarily available in /ccc/scratch/.../IGCM_OUT/IPSLCM5A/DEVT/pdControl/CURFEV13
Script files, Script Outputs and Debug files (if necessary) are available in /ccc/work/.../modips1/config/IPSLCM5_v5/CURFEV13
```

### 2.3.2. run.card at the end of a simulation

At the end of your simulation, the PeriodState parameter of the *run.card* files indicates if the simulation has been **completed** or was aborted due to a **Fatal** error.

This file contains the following sections :

- Configuration : allows you to find out how many integration steps were simulated and what would be the next integration step if the experiment would be continued.

```
[Configuration]
#lastPREFIX
OldPrefix=      # ---> Prefix of the last created files during the simulation = JobName + date of the last period. Use
#Warning : OldPrefix not used anymore from libIGCM_v2.5.
#Compute date of loop
PeriodDateBegin= # --->start date of the next period to be simulated
PeriodDateEnd=   # ---> end date of the next period to be simulated
CumulPeriod=     # ---> number of already simulated periods
# State of Job "Start", "Running", "OnQueue", "Completed"
PeriodState="Completed"

SubmitPath=      # ---> Submission directory
```

- PostProcessing : returns information about the post processing status

```
[PostProcessing]
TimeSeriesRunning=n    # ---> indicates if the timeSeries are running
TimeSeriesCompleted=20091231    # ---> indicates the date of the last TimeSerie produced by the post processing
```

- Log : returns technical (run-time) information such as the size of your executable and the execution time of each integration step.

```
[Log]
# Executables Size
LastExeSize=( )

#-----
# CumulPeriod | PeriodDateBegin | PeriodDateEnd | RunDateBegin | RunDateEnd | RealCpuTime | U
#          1 |      20000101 |      20000131 | 2013-02-15T16:14:15 | 2013-02-15T16:27:34 |      798.33000 |
#          2 |      20000201 |      20000228 | 2013-02-15T16:27:46 | 2013-02-15T16:39:44 |      718.16000 |
```

If the run.card file indicates a problem at the end of the simulation, you can check your Script\_Output file for more details. See [more details here](#).

### 2.3.3. Script\_Output\_JobName

A Script\_Output\_JobName file is created for each job executed. It contains the simulation job output log (list of the executed scripts, management of the I/O scripts).

This file contains mainly three parts :

- copying and handling of input and parameters files
- running the model
- copying of outputs files and launching of post processing steps (rebuild if necessary and pack)

These three parts are defined as below :

```
#####
#          ANOTHER GREAT SIMULATION          #
#####

1st part (copying and handling of the input and parameter files)

#####
#          DIR BEFORE RUN EXECUTION          #
#####

2nd part (running the model)

#####
#          DIR AFTER RUN EXECUTION          #
#####

3rd part (copying of outputs files and launching of post processing steps (rebuild and pack))
```

## 2.4. The output files

The output files are stored on file servers. Their name follows a standardized nomenclature:

IGCM\_OUT/TagName/[SpaceName]/[ExperimentName]/JobName/ in different subdirectories for each "Output" and "Analyse" component (e.g. ATM/Output, ATM/Analyse), DEBUG, RESTART, ATLAS and MONITORING. File servers where outputs will be store depends on the [SpaceName](#) choose for the simulation. Remember : TEST mode is a specific case which deactivate [pack](#)

SpaceName	Computing Center	File Server
TEST	TGCC	\$SCRATCHDIR
TEST	IDRIS	\$SCRATCH

DEVT / PROD	TGCC	\$STOREDIR
DEVT / PROD	IDRIS	\$STORE

Prior to the [packs](#) execution, this directory structure is stored

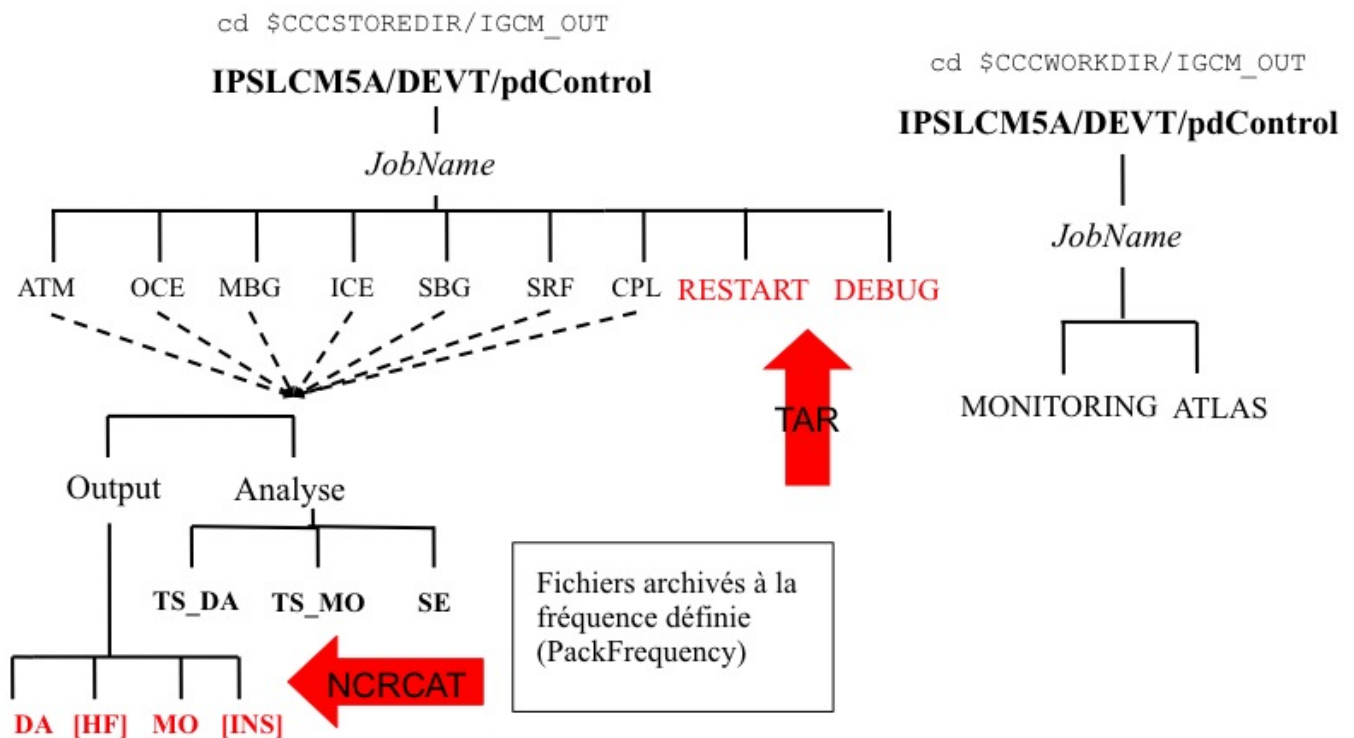
- on the \$CCCSCRATCHDIR at TGCC
- on the \$SCRATCH at IDRIS

After the [packs](#) execution (see diagram below), this tree is stored

- on the \$CCCSTOREDIR and the \$CCCWORKDIR at TGCC
- on the \$STORE at IDRIS

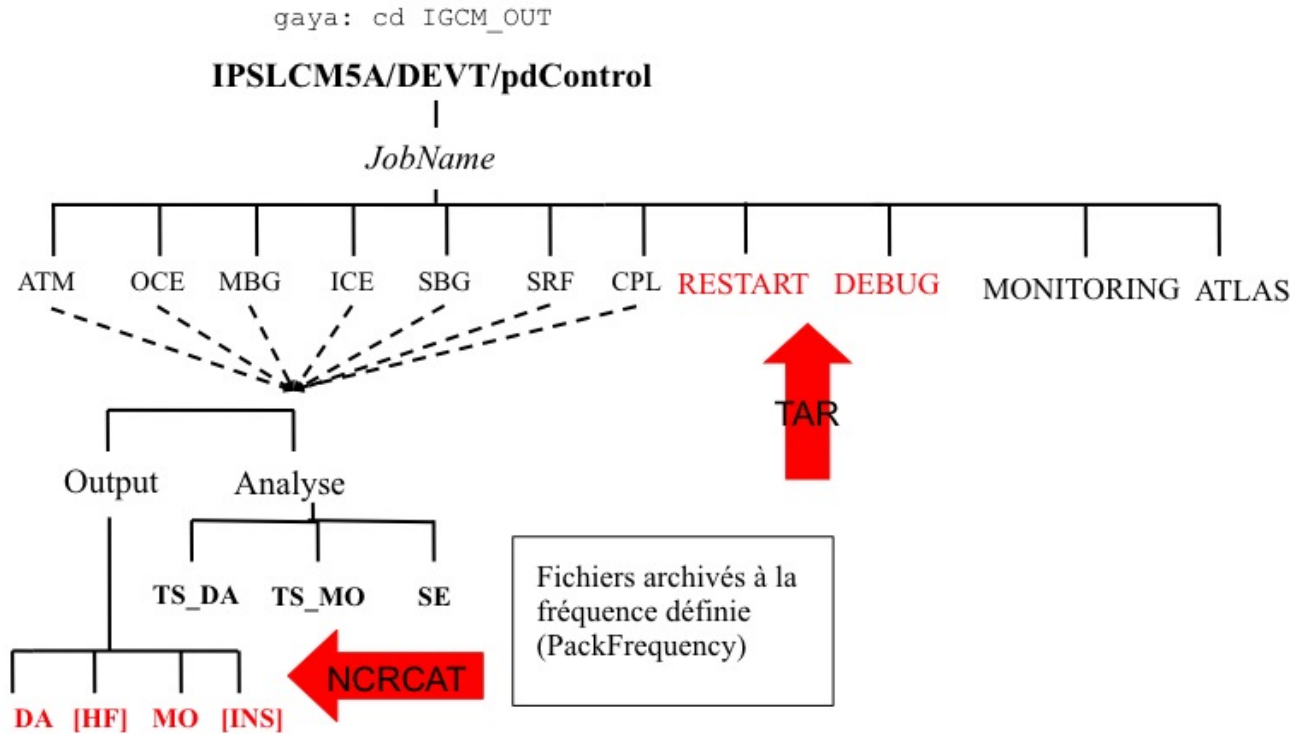
#### 2.4.1. Here is the storage directory structure of the output files produced at TGCC

## Arborescences sur serveurs de fichiers TGCC



#### 2.4.2. Here is the storage directory structure of the output files produced at IDRIS

# Arborescences sur serveur de fichiers IDRIS



## 2.5. Debug/ directory

A Debug/ directory is created if the simulation crashed. This directory contains text files from each of the model components to help you finding reasons for the crash. See also [the chapter on monitoring and debugging](#).

## 2.6. How to continue or restart a simulation?

1. If you want to continue an existing and finished simulation, change the simulation end date in the `config.card` file. Do not change the simulation start date.

In the `run.card` file you must:

- check that the `PeriodDateBegin` and `PeriodDateEnd` variables match with the next integration step of your simulation (e.g. if you just finished May 2000 and you want to integrate one month, set `PeriodDateBegin= 20000601` and `PeriodDateEnd= 20000630`)
  - specify `PeriodState = OnQueue`
3. You must change the output file number in your job to make sure that the job doesn't fail by trying to replace an existing `Script_Output` file. By default it is `Script_Output_JobName_.0001` but you can replace it by `Script_Output_JobName_.CumulPeriod` (you will find `CumulPeriod` in `run.card`)
- If your simulation has stopped in the middle of a month and you want to restart it, you must delete the files created during this month (pack period) in your archives (`$CCCSTOREDIR/IGCM_OUT/etc...`). You can use the scripts [`modips/libIGCM/clean\\_PeriodLength.job`](#) and [`modips/libIGCM/clean\\_latestPackperiod.job`](#).

```
cd $SUBMIT_DIR (ie modips1/config/****/ExpDirectory)
cp ../../../../libIGCM/clean_PeriodLength.job .
./clean_PeriodLength.job # Answer to the questions

same for clean_latestPackperiod.job
```

```
ccc_msub Job_EXP00 or sbatch Job_EXP00
```

### 3. Simulation - Post processing

#### 3.1. Post processing in config.card

You must specify in *config.card* the kind and frequency of the post processing.

```
#=====
#D-- Post -
[Post]
#D- Do we rebuild parallel output, this flag determines
#D- frequency of rebuild submission (use NONE for DRYRUN=3)
RebuildFrequency=NONE
#D- frequency of pack post-treatment : DEBUG, RESTART, Output
PackFrequency=1Y
#D- If you want to produce time series, this flag determines
#D- frequency of post-processing submission (NONE if you don't want)
TimeSeriesFrequency=10Y
#D- If you want to produce seasonal average, this flag determines
#D- the period of this average (NONE if you don't want)
SeasonalFrequency=10Y
#D- Offset for seasonal average first start dates ; same unit as SeasonalFrequency
#D- Useful if you do not want to consider the first X simulation's years
SeasonalFrequencyOffset=0
#=====
```

If no post processing is desired you must specify **NONE** for the TimeSeriesFrequency and SeasonalFrequency frequencies.

#### 3.2. Rebuild

For almost all configurations, the rebuild phase is not needed. Single files are directly written in parallel by XIOS

- `rebuild` is a tool which allows you to combine several files created by a parallel program (sub domains) to a single file. Note that if you use XIOS as output library (XIOS is used in v6 configurations), the rebuild step could not be needed : it depends on the writing mode (parallel or not, server or not) you have activated.
- `rebuild` is available with IOIPSL package. See <http://forge.ipsl.jussieu.fr/igcmg/browser/IOIPSL/trunk/tools> (it can therefore be distributed via modipsl)
- `rebuild` is installed on the IDRIS and TGCC front-end machines.  
`rebuild` is only needed and launched in following cases :
  - IOIPSL is used as output writing library and you run in parallel mode.
  - XIOS is used as output writing library, you run in parallel mode, you run in XIOS attached mode (or server mode with several servers) and you activate `multiple_file` XIOS mode.
- If needed, `rebuild` is automatically called at the `RebuildFrequency` frequency and it is usually the very first step of post processing. Specifying `NONE` for `RebuildFrequency` will start the file combining on the computing machine instead of doing it on the post processing machine. This is strongly discouraged. `RebuildFrequency=1Y` indicates the frequency of running REBUILD. The files to be combined by `rebuild` are stored on a buffer space `$$$SCRATCHDIR/IGCM_OUT/./JobName/REBUILD/` at TGCC and `$WORK/IGCM_OUT/./JobName/REBUILD` at IDRIS. Note: if `JobType=DEV` the parameter is forced to have the `PeriodLength` value.
- `RebuildFromArchive=NONE` is the option to be used on all machines. The REBUILD job first looks for the files to be assembled on the buffer space. Then it assembles them (*rebuild*), applies requested Patches and stores them in the usual `COMP/Output/MO` or `COMP/Output/DA` directories for monthly or daily files of the COMP component (OCE, ICE, ATM, SRF, ...). Note: REBUILD does the ordering of other post processing jobs ran by the `create_ts.job` and `create_se.job` jobs.

#### 3.3. Concatenation of "PACK" outputs



The model outputs are concatenated before being stored on archive servers. The concatenation frequency is set by the **PackFrequency** parameter (NONE means no concatenation, not recommended). If this parameter is not set the rebuild frequency RebuildFrequency is used. This packing step is performed by the PACKRESTART, PACKDEBUG(started by the main job) and PACKOUTPUT (started by the rebuild job or the main job) jobs.

### 3.3.1. How are the different kinds of output files treated ?

All files listed below are archived or concatenated at the same frequency (PackFrequency)

- **Debug** : those files are archived and grouped in a single file with the `tar` command. They are then stored in the `IGCM_OUT/TagName/.../JobName/DEBUG/` directory.
- **Restart** : those files are archived and grouped in a single file with the `tar` command. They are then stored in the `IGCM_OUT/TagName/.../JobName/RESTART/` directory.
- **Output** : those files are concatenated by type (histmth, histday ...) with the `ncrcat` command in the `IGCM_OUT/TagName/.../JobName/_comp_/Output/` directories.

## 3.4. Time Series

A Time Series is a file which contains a single variable over the whole simulation period (ChunkJob2D = NONE) or for a shorter period for 2D (ChunkJob2D = 100Y) or 3D (ChunkJob3D = 50Y) variables.

- The write frequency is defined in the `config.card` file: `TimeSeriesFrequency=10Y` indicates that the time series will be written every 10 years and for 10-year periods.
- The Time Series are set in the `COMP/*`.card files by the `TimeSeriesVars2D` and `TimeSeriesVars3D` options.

Example for `Imdz` :

```

45 [OutputFiles]
46 List= (histmth.nc,      ${R_OUT_ATM_O_M}/${PREFIX}_1M_hismth.nc,      Post_1M_hismth), \
...
53 [Post_1M_hismth]
54 Patches= ( )
55 GatherWithInternal = (lon, lat, presnivs, time_counter, aire)
56 TimeSeriesVars2D = (bils, cldh, ... )
57 ChunkJob2D = NONE
58 TimeSeriesVars3D = ( )
59 ChunkJob3D = NONE

```

- Each output file (section [OutputFiles]) is related to a post processing job: `Post_1M_hismth` in the example.
- `Post_1M_hismth` is a section (starting by "[Post\_1M\_hismth]")
- This section contains the variables : `Patches=` , `GatherWithInternal =` , `TimeSeriesVars2D =` , `ChunkJob2D` , `TimeSeriesVars3D` and `ChunkJob3D`.
  - `Patches= ( )`, this functionality is not used anymore. Before, the patch was applied to the output file. The old patches can be found here: [libIGCM\\_post](#) Different Patches can be applied consecutively.
  - `GatherWithInternal = (lon, lat, presnivs, time_counter, aire)` These are the variables to be extracted from the initial file and to be stored with the Time Series variable.
  - `TimeSeriesVars2D/3D =` those are variable lists of time series to create.
  - `ChunkJob2D/3D =` if the simulation is too long you can cut the time series into x-year long chunks (`ChunkJob2D=50Y` for example).

The Time Series coming from monthly (or daily) output files are stored on the archive server in the `IGCM_OUT/TagName/[SpaceName]/[ExperimentName]/JobName/Composante/Analyse/TS_MO` and `TS_DA` directories.

You can add or remove variables to the TimeSeries lists according to your needs.

There are as many time series jobs as there are `ChunkJob3D` values. This can result in a number of `create_ts` jobs (automatically started by the computing sequence).

## 3.5. Monitoring and intermonitoring

The monitoring is a web-interface tool that visualizes the global mean over time for a set up of key variables. Access the monitoring using the `esgf/thredds` address for your machine ending with `yourlogin/TagName/SpaceName/JobName/MONITORING`. If you have a new account, you might need to contact the assistant team at the computer center to activate your write access to `esgf/thredds`.

- esgf/thredds at TGCC:
  - [https://vesg.ipsl.upmc.fr/thredds/catalog/work\\_thredds/YOURLOGIN/catalog.html](https://vesg.ipsl.upmc.fr/thredds/catalog/work_thredds/YOURLOGIN/catalog.html)
- esgf/thredds at IDRIS:
  - to be completed...

The key variables plotted in the monitoring are computed using Time Series values. The monitoring is updated at the TimeSerieFrequency set in *config.card* if the time series were successfully done. This allows you to monitor a simulation. By monitoring your simulations you can detect anomalies and evaluate the impact of changes you have made. We suggest to create a tab in your browser allowing you to frequently monitor your simulation. If a few key variables start looking suspicious you might want to stop your simulation. By doing so, you will save computing time. A full documentation is available at <http://wiki.ipsl.jussieu.fr/IGCMG/Outils/ferret/Monitoring>.

Here is an example for the IPSLCM5A coupled model and a 10-year period. Once you are in yourlogin/TagName/SpaceName/JobName/MONITORING, you have to click on index.html. The first tab called **Analysis Cards** gives a summary of dates and execution times obtained from the *config.card* and *run.card* files. The second tab called **Monitoring Board** presents a monitoring table for the key variables (selecting one or more model components is optional).

## BAL1210 monitoring

at 2010-12-06 05:38:15

Cards Analysis
Monitoring Board
About

- Progress of the simulation
- run.card
  - [Open at original format](#)
  - [Open as sortable table](#)
- config.card
  - [Open at original format](#)
- Directories access
  - [files](#)
  - [images](#)
  - [progress](#)

- Simulation date summary

CalendarType	DateBegin	DateEnd	DateCurrent	PeriodLength	CumulPeriod
noleap	2000-01-01	2009-12-31	2010-01-31	1M	121
- Real Cpu time summary

min	max	average
967.67	1086.38	1050.64
- User Cpu time summary

min	max	average
0.94	3.28	1.20

**BAL1210 monitoring**

at 2010-12-06 05:38:15

Cards Analysis Monitoring Board About

ALL

Filter : ICE

Images : 010 / 067

ATM

CHM

ICE

MBG

OCE

SBG

SRF

XOR

CLR

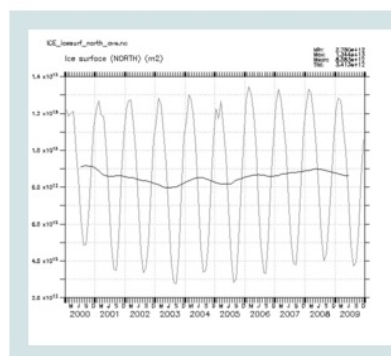
land

ocean

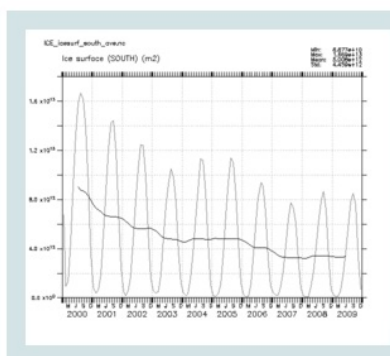
north

south

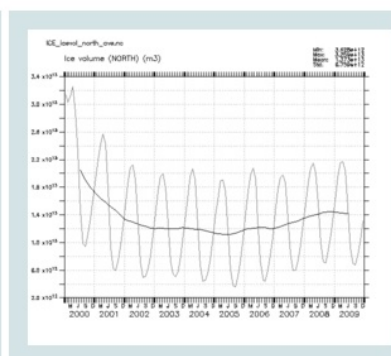
global



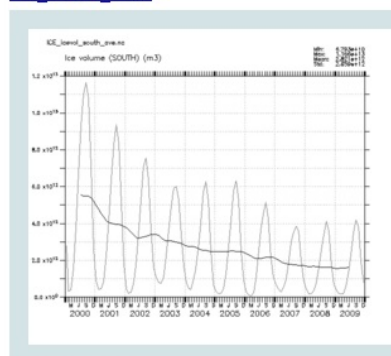
ICE\_icesurf\_north



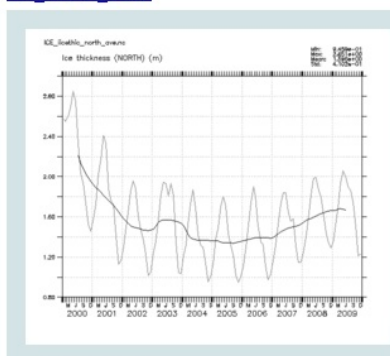
ICE\_icesurf\_south



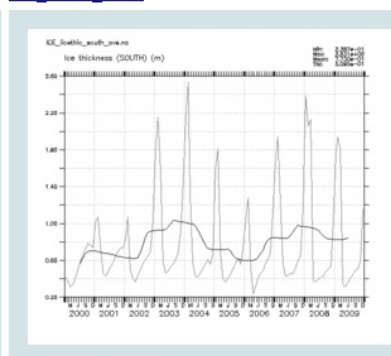
ICE\_icevol\_north



ICE\_icevol\_south



ICE\_icethic\_north



ICE\_icethic\_south



- The diagnostics of each experiment are stored in the MONITORING directory following the IGCN\_OUT/TagName/SpaceName/ExperimentName/MONITORING nomenclature (on the \$CCCWORKDIR at TGCC and on \$WORK at IDRIS (to be modified...)).
- The diagnostics starts automatically after the Time Series are created. See the diagram on the computing sequence.

**3.5.1. Adding a variable to the monitoring**

You can add or change the variables to be monitored by editing the configuration files of the monitoring. Those files are defined by default for each component.

By default, the monitoring is defined here: `~compte_commun/atlas` For example for LMDZ: `monitoring01_lmdz_LMD9695.cfg`

You can change the monitoring by creating a `POST` directory which is part of your configuration. Copy a `.cfg` file and change it the way you want. You will find two examples in [special post processing](#)

**Be careful :** to calculate a variable from two variables you must define it within parenthesis :

```
#-----
# field | files patterns | files additionnal | operations | title | units | calcul of area
#-----
nettop_global | "tops topl" | "" | "(tops[d=1]-topl[d=2])" | "TOA. total heat flux (GLOBAL)" | "W/m2" | "aire[d=1]"
tops_global | "tops" | "" | "tops[d=1]" | "tops Global" | "W/m2" | "aire[d=1]"
```

- field can not have the same name than the variable used for the monitoring (--> it creates an error) (tops\_global / tops in the example)
- units need to be complete (if you have "" for units it creates an error --> you need to indicate one character)
- if "aire" or "area" is not defined in the TS file, you can add an additional file to find this variable (and in this case it changes the number of files for the last parameter)

```
#-----
# field | files patterns | files additionnal | operations | title | units | calcul of area
#-----
nettop_global | "tops topl" | LMDZ4.0_9695_grid.nc | "(tops[d=1]-topl[d=2])" | "TOA. total heat flux (
```

by default monitoring is based on TS\_MO files, but it is possible to change the frequency by adding **FreqTS** in the configuration file monitoring\*.cfg

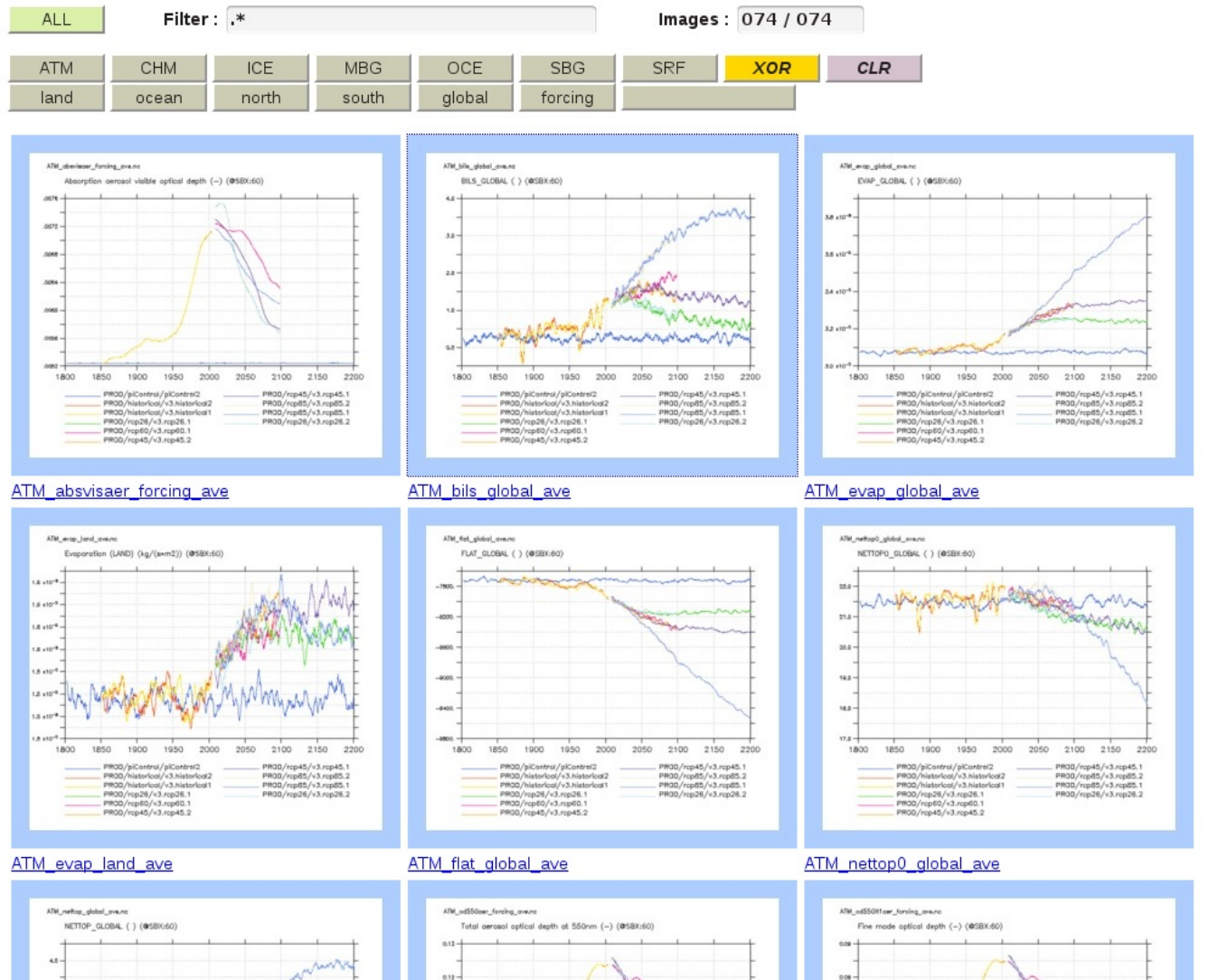
```
FreqTS=DA
#-----
# field | files patterns | files additionnal | operations | title | units | calcul of area
#-----
nettop_global | "tops topl" | LMDZ4.0_9695_grid.nc | "(tops[d=1]-topl[d=2])" | "TOA. total heat flux (
```

### 3.5.2. Inter Monitoring

- To simultaneously monitor various simulations, a web application [Intermonitoring](#) has been created to monitor the evolution of different variables at once.

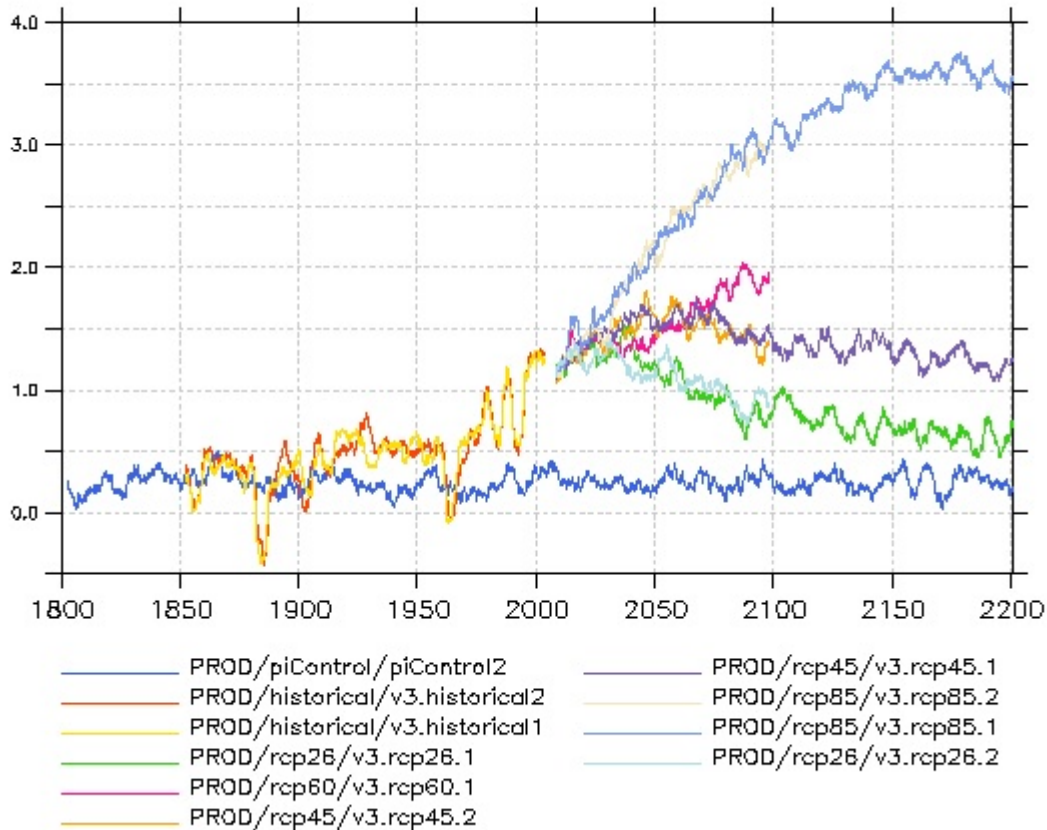
# Monitoring comparison: piControl2 vs v3.historical2 vs v3.historical1 vs v3.rcp26.1 vs v3.rcp60.1 vs v3.rcp45.2 vs v3.rcp45.1 vs v3.rcp85.2 vs v3.rcp85.1 vs v3.rcp26.2

at 2011-05-29 21:49:27



ATM\_bile\_global\_ave.nc

BILS\_GLOBAL ( ) (@SBX:60)



- This application is available on [ICMC Web Applications / intermonitoring](#).
  - A video tutorial is available [here](#).
- In the **Step 1** tab, specify URLs which give access to the MONITORING directory
- to be completed... at IDRIS
  - [http://vesg.ipsl.upmc.fr/thredds/catalog/work\\_thredds/YourLogin/TagName/SpaceName/ExperimentName](http://vesg.ipsl.upmc.fr/thredds/catalog/work_thredds/YourLogin/TagName/SpaceName/ExperimentName) at TGCC (WORKDIR directory)

### 3.5.3. Mini how to use the intermonitoring

Go to <http://webservices.ipsl.fr/interMonitoring>

- Step 1: Enter the first path and click on the button List Directories.
- Step 2: You'll see a list of all simulations at this path. Go back to step 1.
- Step 1 bis: Enter the second path and click on Append directories.
- Step 2 bis: You'll now see all simulations on the 2 paths. Choose the two or more simulations (use the mouse and type ctrl to select only 2 simulations). Click on Search files.
- Step 3: Select one variable and click on Validate.
- Step 4: Choose default setting "plot01:Time series" and click on Validate. Then click on the button below called Prepare and run the ferret script.
- Step 5: Now a ferret script will appear on the screen and one image. Click on the button Run this script on the server below on the page. The inter-monitoring for all variables will now appear on the screen.

### 3.5.4. How to save the intermonitoring permanently

The plots done by the intermonitoring will be kept 30 days. During these days you can visualize using the same link the plots done. To keep them permanently, do as follow:

- Create the intermonitoring using the webservices interface (see mini howto or audio guide above)

- Save the .jnl script and the .bash script created by the webservises to your computer, together in the same directory.
- Edit the .bash and modify as follows :
  - source of ferret configuration files. Here are examples to uncomment if needed :

```
# # IPSL (webservices)
# # IDRIS (jeanzay)
#. /home/users/brock/.atlas_env_asterix_bash      # LSCE (asterix)
# # IPSL (ciclad)

# add for TGCC (irene)
module unload ferret
module load pyferret/7.4
module load imagemagick/6.9.1-2
. ~/igcmg/MachineEnvironment/irene/env_atlas_irene

# add for ciclad:
. /home/brocksce/.atlas_env_cicladl_bash
```

- define the name and the path where you saved the .jnl script.

```
scriptname=./intermonit_CM6.jnl
```

- uncomment and adapt the last line in the script with copy to esgf/thredds. For example at ciclad :

```
# for ciclad
cp -fR ${scriptname%%.jnl}_prod /prodigfs/ipslfs/http/dods/web/html/login/INTERMONITORING

# or for irene:
cp -fR ${scriptname%%.jnl}_prod $CCCWORKDIR/../../thredds/login/INTERMONITORING
```

- For irene you need to edit the .jnl script as follows:
  - change the path of the netCDF file to plot, inserting the SpaceName, ExperimentName, JobName related to your run:

```
use "$CCCWORKDIR/../../thredds/login/TagName/SpaceName/ExperimentName/JobName/MONITORING/files/($FILE)"
```

- change permissions on .bash

```
chmod 755 *.bash
```

- Run the .bash script. A new directory will appear on your computer. This is the directory that is also copied to esgf/thredds.
- The intermonitoring is now on esgf/thredds and you can keep the link permanently. For example for irene, you can find intermonitoring following the link below, where XXXXX is the name of the .jnl file (without .jnl) :

```
http://vesg.ipsl.upmc.fr/thredds/fileServer/work/login/INTERMONITORING/XXXXX_prod/index.html
```

### 3.6. Seasonal means

The SE (seasonal means) files contain averages for each month of the year (jan, feb,...) for a frequency defined in the *config.card* files

- SeasonalFrequency=10Y The seasonal means will be computed every 10 years.
- SeasonalFrequencyOffset=0 The number of years to be skipped for calculating seasonal means.
- The SE files will automatically start at the SeasonalFrequency=10Y frequency (pay attention to SeasonalFrequencyOffset=0) when the last parameter in the file of the '[OutputFiles]' section is not NONE.
- All files with a requested Post are then averaged within the ncra script before being stored in the directory:  
IGCM\_OUT/IPSCLCM5A/DEVT/pdControl/MyExp/ATM/Analyse/SE. There is one file per SeasonalFrequency=10Y

### 3.7. Atlas



- The atlas is a result of the automatic post processing, which will create a collection of plots presented as a web tree. Each plot is available as image and pdf file. The plots are made with the software ferret and the FAST and ATLAS libraries. More information on ferret and on those libraries can be found here <http://wiki.ipsl.jussieu.fr/IGCMG/Outils/ferret/Atlas>
- By default, automatic atlas are disabled. To activate automatic atlas, you have to specify `config_Post_AtlasIPSL=TRUE` in [Post part](#) of `config.card`
- Here is an example of atlas for the coupled IPSLCM5A available on esgf/thredds : [ATM](#)
- There are at least 8 directories with Atlas for the coupled model. They are based on `atlas_composante.cfg` files. You can look at them on the file servers in the `IGCM_OUT/IPSLCM5A/DEVT/pdControl/MyExp/ATLAS` directories.
- The script libraries `fast/atlas` are installed in shared directories on every machine.
- Note that it is also possible to generate CLIMAF atlas : you have to launch it on your own. See [documentation](#)

### 3.8. Storing files like ATLAS, MONITORING and ANALYSE

The files produced by ATLAS, MONITORING, Time series and Seasonal means are stored in the following directories:

- **ANALYSE**: copied to `JobName/_comp_/Analyse` on the file server **store** or **ergon** (to be modified)
- **MONITORING**: copied to `JobName/MONITORING` on the file server **work** or **ergon** (to be modified)
- **ATLAS**: copied to `JobName/Atlas` on the file server **work** or **ergon** (to be modified)

They are available through esgf/thredds server at IDRIS and at TGCC.

- at TGCC with the new one called esgf/thredds:
  - <https://vesg.ipsl.upmc.fr/thredds/catalog/catalog.html> click on `work_thredds`, click on your login and... for ATLAS and MONITORING
  - <https://vesg.ipsl.upmc.fr/thredds/catalog/catalog.html> click on `store_thredds`, click on your login and... for Analyse files (TS or SE)
- at IDRIS: <https://prodn.idris.fr/thredds>

### 3.9. How to check that the post processing was successful

Since 2016, `hermes.ipsl.upmc.fr` is recommended to check and follow your simulations (computing and post-processing jobs). More information [here](#). Besides, the post processing output log files are :

- on [JeanZay?](#): `$WORK/IGCM_OUT/TagName/.../JobName/Out`
- on Irene: `$CCSCRATCHDIR/IGCM_OUT/TagName/.../JobName/Out`

In these directories, you find the job output files for following jobs: `rebuild`, `pack*`, `ts`, `se`, `atlas`, `monitoring` .

Scripts to transfer data on esgf/thredds are run at the end of the monitoring job or at the end of each atlas job.