Wikiprint Book

Title: 1. Submitting your simulation

Subject: lgcmg_doc - Doc/Running

Version: 105

Date: 07/03/24 06:45:26

Table of Content

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 of the outputs.

1. Submitting your simulation

Once you have defined and setup your simulation you can submit it. The run scripts are:

- ccc_msub at TGCC
- llsubmit at IDRIS

```
curie > ccc_msub Job_MYJOBNAME
ada   > llsubmit Job_MYJOBNAME
```

These scripts return a job number that can be used with the machine specificities to manage your job. Please refer to the environment 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 chain of jobs, depending on the kind of jobs you submitted. These jobs include the production job itself and post processing jobs like: rebuild, pack, create_ts, create_se, monitoring and atlas which are started at given frequencies.

Post processing jobs will be automatically started at the end of the simulation.

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

2. Status of the running simulation

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. Execution directory

- At TGCC your simulation is performed in a $SCRATCHDIR/RUN_DIR/job_number directory. You can check the status of your simulation in this directory.
- At IDRIS your simulation is performed in a temporary directory. You must first specify **RUN_DIR_PATH=$WORKDIR** in your production job if you want to monitor it.

3. End of the simulation

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 curie3820.
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/.../modipsl/config/IPSLCM5_v5/CURFE
```

Example of message when the simulation failed

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

Dear Jessica,


Simulation CURFEV13 is failed on supercomputer curie3424.
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/.../modipsl/config/IPSLCM5_v5/CURFE
```

3.2. How to continue or restart a simulation

1. If you want to continue an existing or 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= 2000630`)
   - 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_NomJob_.0001` but you can replace it by `Script_Output_NomJob_.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 `modipsl/libIGCM/clean_month.job` and `modipsl/libIGCM/clean_year.job`.

   ```
   cd $SUBMIT_DIR (ie modipsl/config/LMDZOR_v5/DIADEME)
   cp ../../../libIGCM/clean_month.job . ; chmod 755 clean_month.job  # Once and for all
   ./clean_month.job  #   Answer to the questions

   same for clean_year.job

   ccc_msub Job_EXP00 or llsubmit Job_EXP00
   ```

3.3. 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.

Prior to the packs execution, this directory structure is stored

- on the $SCRATCHDIR at TGCC
- on the $WORKDIR at IDRIS

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

- on the $CCCSTOREDIR and the $CCCWORKDIR at TGCC
- on the Ergon machine at IDRIS

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

Arborescences sur serveurs de fichiers TGCC

cd $CCCSTOREDIR/IGCM_OUT

IPSLCM5A/DEVT/pdControl

cd $CCCWORKDIR/IGCM_OUT

IPSLCM5A/DEVT/pdControl

JobName

ATM OCE MBG ICE SBG SRF CPL RESTART DEBUG

JobName

MONITORING ATLAS

TAR

Output   Analyse

TS_DA   TS_MO   SE

Fichiers archivés à la fréquence définie (PackFrequency)

NCRCAT

DA [HF] MO [INS]
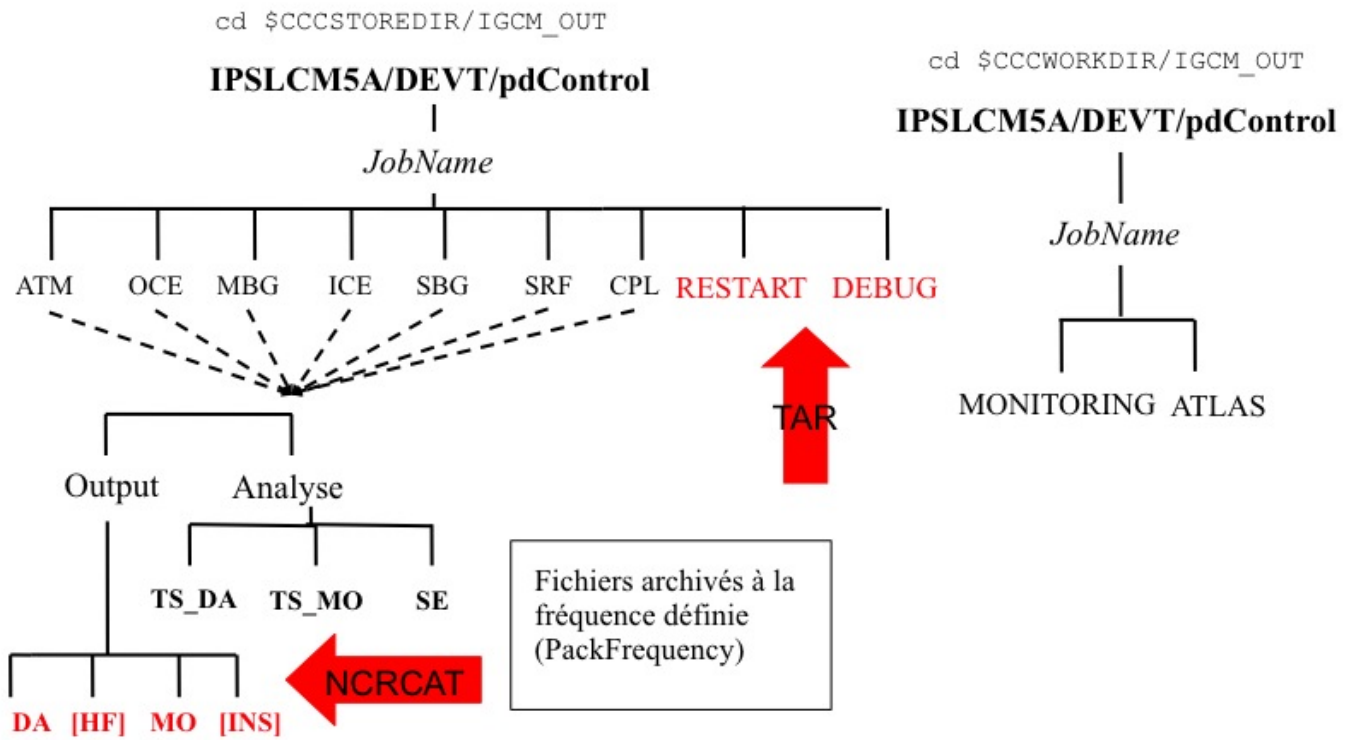
3.3.2. Here is the storage directory structure of the output files produced at IDRIS

Arborescences sur serveur de fichiers IDRIS

```
gaya: cd IGCM_OUT
```

IPSLCM5A/DEVT/pdControl

JobName

ATM   OCE   MBG   ICE   SBG   SRF   CPL   RESTART   DEBUG   MONITORING   ATLAS

TAR

Output   Analyse

TS_DA   TS_MO   SE

Fichiers archivés à la
fréquence définie
(PackFrequency)

NCRCAT

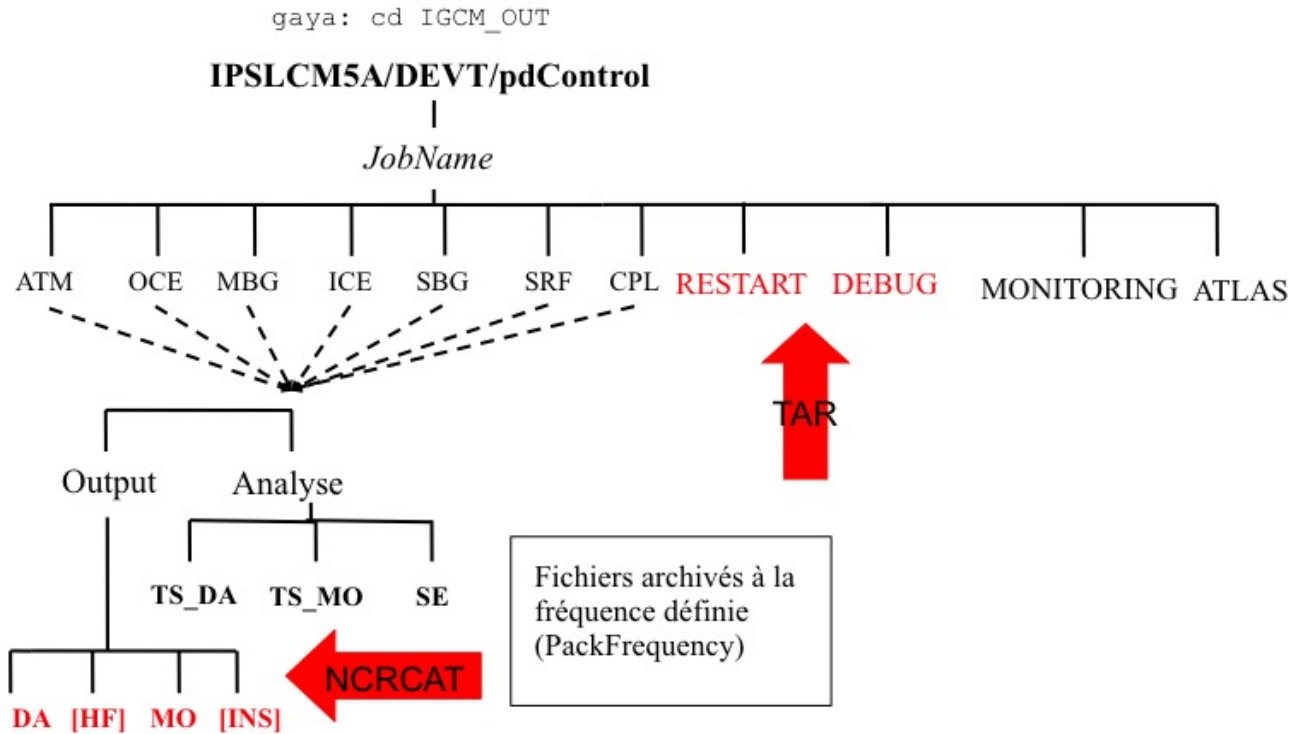DA   [HF]   MO   [INS]

3.4. 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 files 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]
#last PREFIX
OldPrefix=        # ---> prefix of the last created files during the simulation = JobName + date of the last period. Use
#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 |    Us
#           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 |
```

3.5. 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 three parts :

- copying the input files
- running the model
- post processing

These three parts are defined as below :

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

1st part (copying the input files)


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

2nd part (running the model)


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

3rd part (post processing)
```

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.

3.6. Debug/ directory

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

---

4. Simplified diagram of the jobs executed by the IPSL models and tools

The job automatically runs post processing jobs during the simulation and at different frequencies. There are two kinds of post processing: required post processing such as rebuild and pack, and optional post processing such as TimeSeries and seasonal means.
Here is a diagram describing the job sequence.
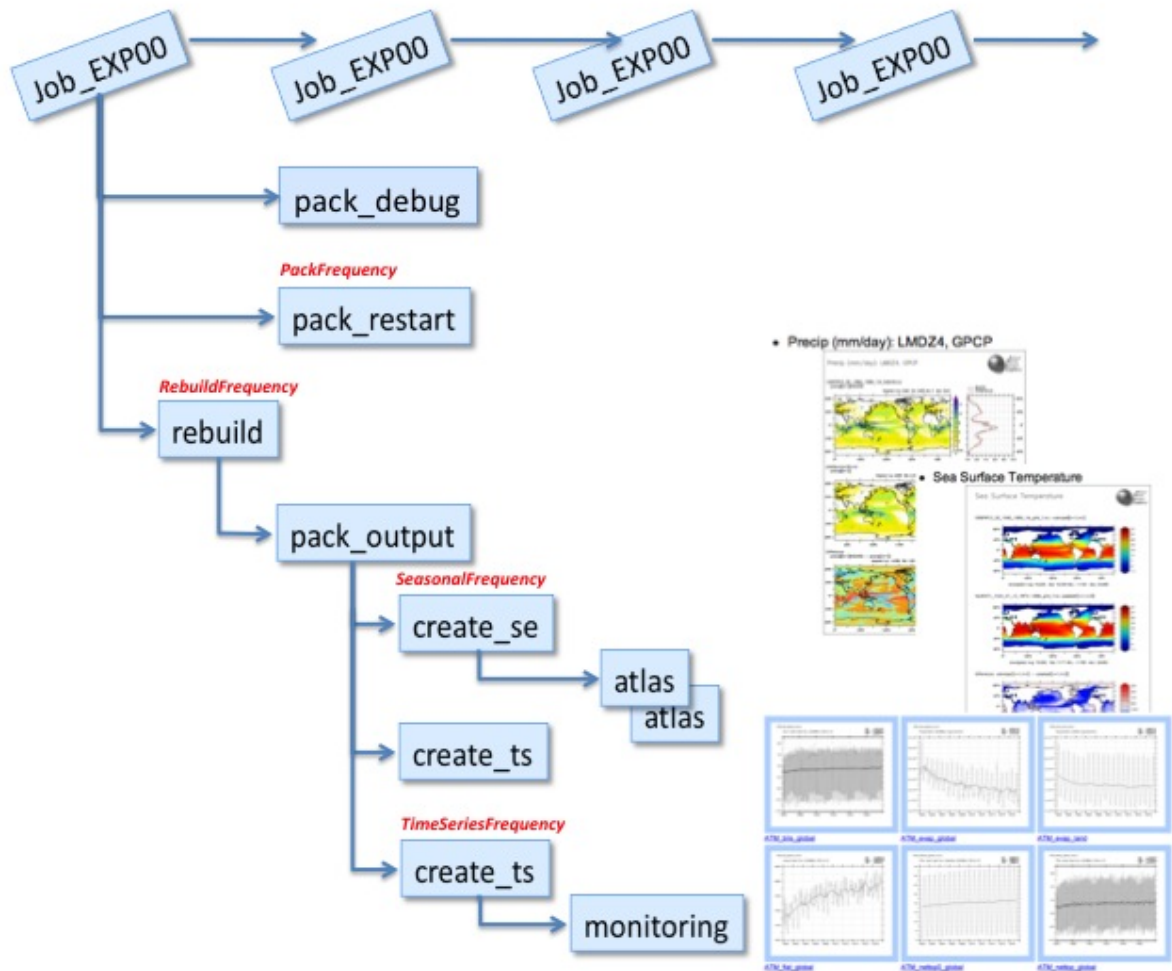
5. Post processing with libIGCM

5.1. 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=1Y
#D- frequency of pack post-treatment : DEBUG, RESTART, Output
PackFrequency=1Y
#D- Do we rebuild parallel output from archive (use NONE to use SCRATCHDIR as buffer)
RebuildFromArchive=NONE
#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- Usefull 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.

5.2. Rebuild

- rebuild is a tool which allows you to combine several files created by a parallel program (sub domains) to a single file.
- rebuild is available with IOIPSL. 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. It is automatically called at the RebuildFrequency frequency and it is usually the very first step of post processing.
- You can not skip rebuilds. 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 $WORKDIR/IGCM_OUT/.../JobName/REBUILD at IDRIS (in old libIGCM version, before libigcm_v2.0, it was $SCRATCHDIR/REBUILD/ at TGCC and $WORKDIR/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 Patchs 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.

Note: if JobType=DEV, the RebuildFrequency parameter is forced to be the PeriodLength value and one rebuild job per simulated period is started. Discouraged for long simulations.

5.3. Concatenation of "PACK" outputs

The model outputs are concatenated before being stored on file servers. The concatenation frequency is set by the **PackFrequency** parameter. If this parameter is not set the rebuild frequency RebuildFrequency is used.
This packing step is performed by the PACKRESTART, PACKDEBUG jobs (started by the main job) and PACKOUTPUT (started by the Rebuild job).

5.3.1. 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.

5.4. Time Series

A Time Series is a file which contains a single variable over the whole simulation period (!ChunckJob2D = NONE) or for a shorter period for 2D (!ChunckJob2D = 100Y) or 3D (!ChunckJob3D = 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 lmdz :

```
45  [OutputFiles]
46  List=   (histmth.nc,      ${R_OUT_ATM_O_M}/${PREFIX}_1M_histmth.nc,      Post_1M_histmth), \
...
53  [Post_1M_histmth]
54  Patches= (Patch_20091030_histcom_time_axis)
55  GatherWithInternal = (lon, lat, presnivs, time_counter, aire)
56  TimeSeriesVars2D = (bils, cldh, ... )
```

```
57   ChunckJob2D = NONE
58   TimeSeriesVars3D = ()
59   ChunckJob3D = NONE
```

- Each output file (section [OutputFiles]) is related to a post processing job: Post_1M_histmth in the example.
- post_1M_histmth is a section (starting by "[Post_1M_histmth]")
- This section contains the variables : Patches= , GatherWithInternal = , !TimeSeriesVars2D = , !ChunckJob2D , !TimeSeriesVars3D and !ChunckJob3D.
  - Patches= (Patch_20091030_histcom_time_axis) : this is the Patch which will be applied to the output file before being stored on the file server. The available 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.
  - !ChunckJob2D/3D = if the simulation is too long you can cut the time series into x-year long chunks (!ChunckJob2D=50Y for example).

The Time Series coming from monthly (or daily) output files are stored on the file 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 !ChunckJob3D values. This can result in a number of create_ts jobs (automatically started by the computing sequence).

5.5. Monitoring

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

- dods at curie: ■http://dods.extra.cea.fr/work
- dods at ada: ■http://dodsp.idris.fr

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 and to check the status during a en going simulation and afterwards. 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.
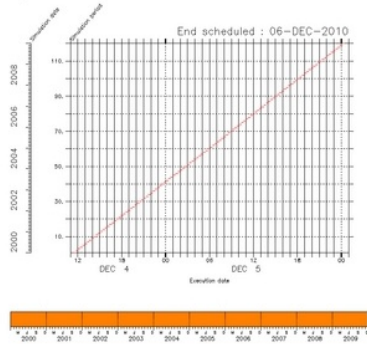
Here is an example for the IPSLCM5A coupled model and a 10-year period. The first tab called **Analysis Cards** gives a summary of dates and execution times obtained from the config.card and run.card files. The second tab called **Monitoring Board** presents a monitoring table for the key variables (selecting 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

- 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

- run.card
  - Open at original format
  - Open as sortable table

- config.card
  - Open at original format

- Directories access
  - files
  - images
  - progress

BAL1210 monitoring

at 2010-12-06 05:38:15
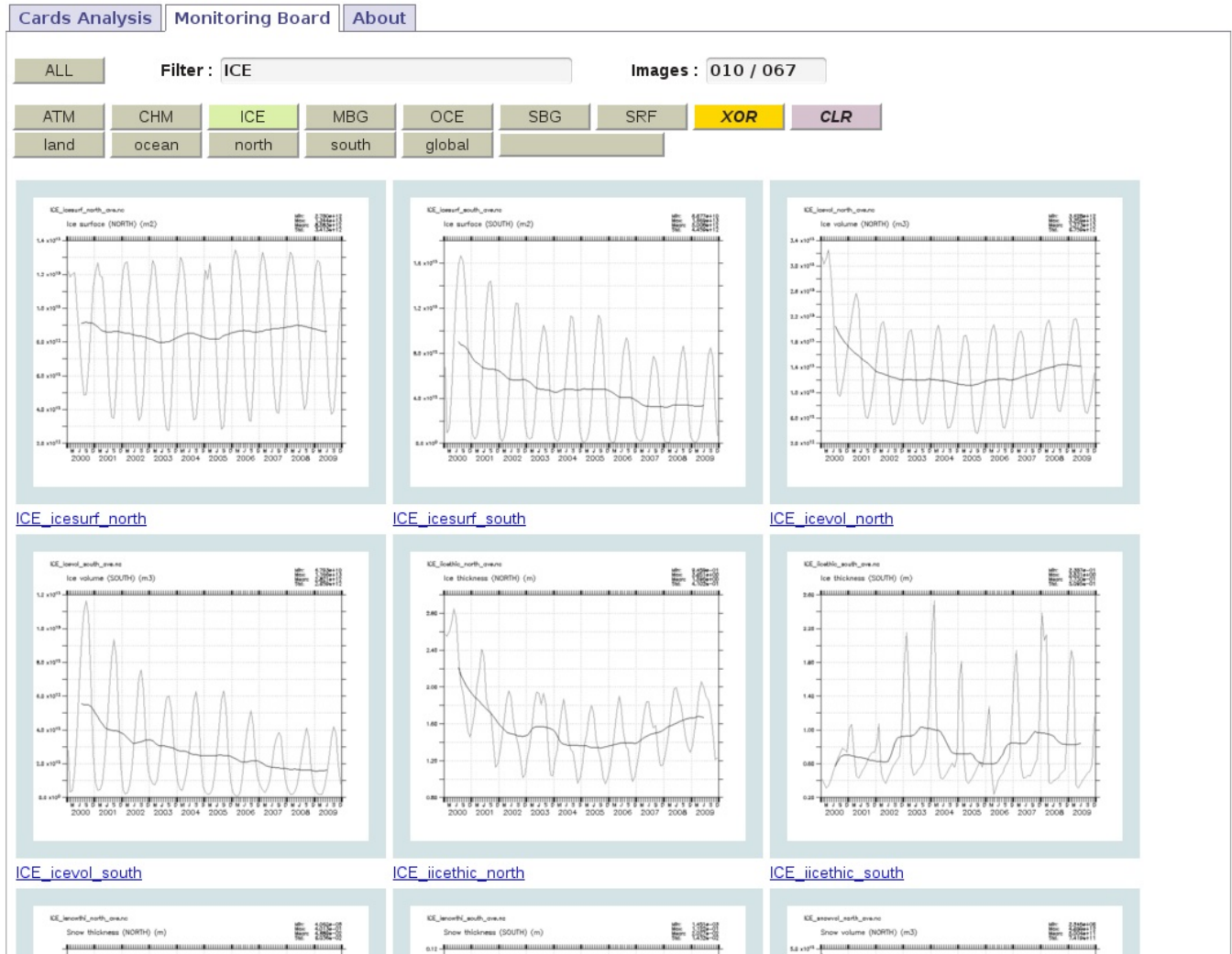
- The diagnostics of each experiment are stored in the MONITORING directory following the IGCM_OUT/TagName/SpaceName/ExperimentName/MONITORING nomenclature (on the $CCCWORKDIR at TGCC and on ERGON at IDRIS).
- The diagnostics starts automatically after the Time Series are created. See the diagram on the computing sequence.

5.5.1. 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.

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"                    | LMDZ4.0_9695_grid.nc | "(tops[d=1]-topl[d=2])" | "TOA. total heat flux (GLO
```
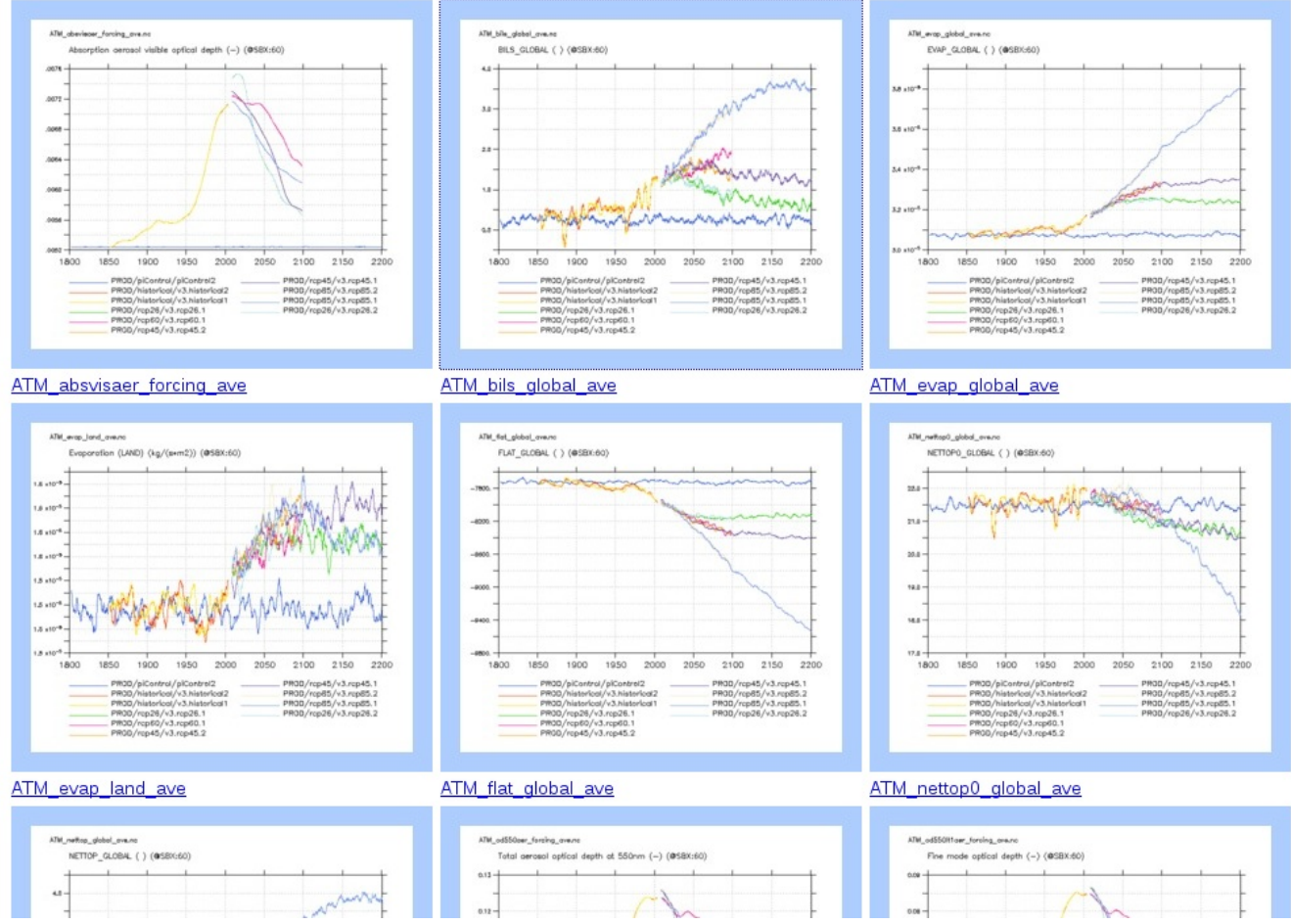
5.5.2. Inter Monitoring

- To simultaneously monitor various simulations, a web application 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
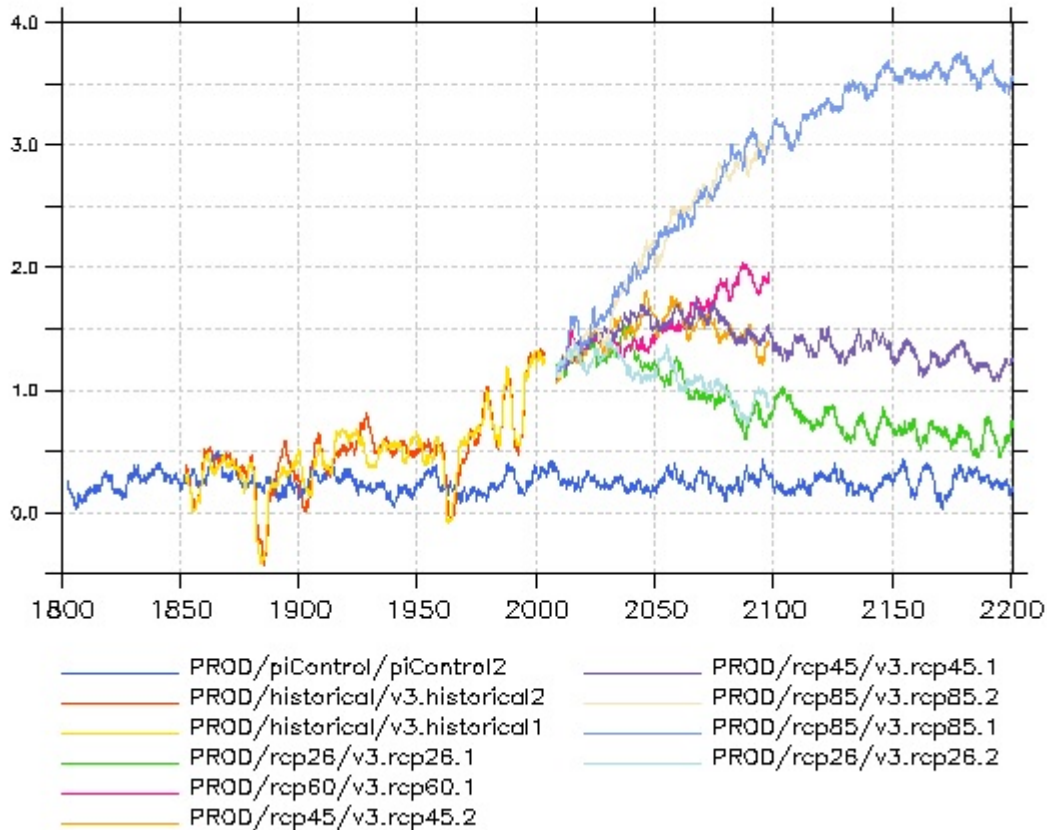
at 2011-05-29 21:49:27

ATM_absvisaer_forcing_ave

ATM_bils_global_ave

ATM_evap_global_ave

ATM_evap_land_ave

ATM_flat_global_ave

ATM_nettop0_global_ave

ATM_bils_global_ava.nc

BILS_GLOBAL ( ) (@SBX:60)

Legend:
- PROD/piControl/piControl2
- PROD/historical/v3.historical2
- PROD/historical/v3.historical1
- PROD/rcp26/v3.rcp26.1
- PROD/rcp60/v3.rcp60.1
- PROD/rcp45/v3.rcp45.2
- PROD/rcp45/v3.rcp45.1
- PROD/rcp85/v3.rcp85.2
- PROD/rcp85/v3.rcp85.1
- PROD/rcp26/v3.rcp26.2

- This application is available on ■**ICMC Web Applications**.
- A video tutorial is available ■here.

  In the **Step 1** tab, specify URLs which give access to the MONITORING directory

  - ■http://dods.extra.cea.fr/cgi-bin/nph-dods/work/YourLogin/TagName/SpaceName/ExperimentName for dods under cccwork at TGCC
  - ■http://dods.extra.cea.fr/cgi-bin/nph-dods/data/YourLogin/TagName/SpaceName/ExperimentName for dods under dmnfs at TGCC (read-only since April 2012)
  - ■http://dods.idris.fr/cgi-bin/nph-dods/YourLogin/TagName/SpaceName/ExperimentName for dods at IDRIS

5.6. 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/IPSLCM5A/DEVT/pdControl/MyExp/ATM/Analyse/SE. There is one file per SeasonalFrequency=10Y

5.7. Atlas

- The atlas is a result of the 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
- Here is an example of atlas for the coupled IPSLCM5A available on dods : ■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.

5.8. Storing files like ATLAS, MONITORING and ANALYSE

The files produced by ATLAS, MONITORING, time series and seasonal means are stored in the directories:

- **ANALYSE**: copied to JobName/_comp_/Analyse on the file server **cccstore** or **ergon**
- **MONITORING**: copied to JobName/MONITORING on the file server **cccwork** or **ergon**
- **ATLAS**: copied to JobName/Atlas on the file server **cccwork** or **ergon**

6. How to check that the post processing was successful

The post processing output log files are :

- on Ada: `$WORKDIR/IGCM_OUT/TagName/.../JobName/Out`
- on Curie: `$SCRATCHDIR/IGCM_OUT/TagName/.../JobName/Out`

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

The scripts to transfer data on dods are run at the end of the monitoring job or at the end of each atlas job.