Wikiprint Book

Title: 1. Create submission directory

Subject: Igcmg_doc - Doc/Setup

Version: 104

Date: 06/29/24 11:11:13

Table of Content

Simulation setup

This chapter describes how to setup your simulation.

1. Create submission directory

The configuration directory contains tools to compile (Makefile and AA_make) and tools to run a simulation, i.e.

- two directories (EXPERIMENTS and GENERAL) allow you to create submission directories for your model configuration;
- if one or several submission directories (e.g. EXP00, OOL_SEC_STO, historical, EXP_AER, etc...) have already been created, you can directly go to the next step.

In the EXPERIMENTS directory you will find subdirectories for each model configuration (included the one you work with). For example:

- IPSLCM5_v5 includes LMDZOR, LMDZ
- IPSLCM5CHT_v5 includes IPSLCM5, LMDZOR, LMDZORINCA
- LMDZOR_v5 includes LMDZ
- LMDZORINCA_v5 includes LMDZOR, LMDZ

Each of these subdirectories may contain a reference experiment (e.g. **clim** and **amip** for LMDZOR, NMHC_AER, AER and GES for LMDZORINCA, **piControl**, **historical** and cie for IPSLCM5_v5) and the file *config.card* which will be your simulation's initial setup.

Before preparing the working directory you must know which kind of simulation you want to perform. Then, you must copy the *config.card* file at the same level as the main *Makefile*.
For example, to perform a **clim** experiment with LMDZOR_v5:

```
cd modipsl/config/LMDZOR_v5
cp EXPERIMENTS/LMDZOR/clim/config.card .
```

The header of *config.card* contains the **JobName** field for which you must specify your simulation's name. Then run the **`ins_job`** script that will in first time ask you, if you are working on TGCC, your id group, and then create a directory for your experiment. If you are working on IDRIS it will directly create a directory for your experiment.
In the previous example, a simulation called *DIADEME* is created:

```
cd modipsl/config/LMDZOR_v5
cp EXPERIMENTS/LMDZOR/clim/config.card .
ls
 AA_Make Makefile EXPERIMENTS GENERAL config.card
vi config.card      # Change JobName=DIADEME
                   # Change [Executable] section to specify your executionType. see config.card description
../../util/ins_job  # Check and complet job's header
ls
 AA_Make Makefile EXPERIMENTS GENERAL DIADEME
```

A directory called *DIADEME* is created and the config.card file is moved in.

2. Contents of the submission directory

The contents of the new directory are described below.

```
cd DIADEME
ls
 config.card COMP/ PARAM/ POST/ DRIVER/
```

2.1. config.card

The *config.card* file contains the settings of your simulation configuration. The file contains several sections with the simulation settings (e.g. name, duration, processors' number, post processing, initial state).
Below is a list of the file sections:

2.1.1. The [UserChoices] section

- JobName --> simulation name
- ExperimentName --> experiment name (following the CMIP5 nomenclature for the IPCC simulations)
- SpaceName --> variable indicating the type of a simulation. Choose between PROD, DEVT and TEST. SpaceName=TEST is a special case deactivating pack and storage.
- LongName --> description of your simulation
- TagName --> **do not change this field**; describes to which configuration family your experiment belongs
- ExpType --> **do not change this field**; allows you to find the EXPERIMENTS directory in which you are working
- DateBegin --> simulation start date (yyyy-mm-dd)
- DateEnd --> simulation end date. It must be the last day "included" in your simulation
- PeriodLength --> frequency of the executable run. This parameter can be 1M, 1Y or 10Y
- JobNumProcTot --> number of processors required by your simulation. This parameter depends on the configurations and on the resolutions. For example, a configuration containing LMDZ in 96x95 can use 32 processors at the most.
- ARCHIVE --> optional : path to base directory for output files. By default this is set in libIGCM depending on the machine.

The parameters ExperimentName and SpaceName are optional. They impact on the path to the storage directory for the simulation output. SpaceName=TEST is a specific case which deactivate pack and storage at archive directory which means that the output will be stored only at SCRATCHDIR(curie) or WORKDIR(ada).

Example 1: The output directory for the following case will be IGCM_OUT/LMDZOR/TEST/REINE/DIADEME

```
JobName=DIADEME
ExperimentName=REINE
SpaceName=TEST
TagName=LMDZOR
```

The output directory will be IGCM_OUT/LMDZOR/TEST/REINE/DIADEME

Example 2: without ExperimentName and SpaceName

```
JobName=DIADEME
TagName=LMDZOR
```

The output directory will be IGCM_OUT/LMDZOR/DIADEME

The character "_" is not allowed in the variables JobName, ExperimentName and SpaceName

PeriodLength allows you to determine the integration length of an execution for your configuration (restart files creation frequency)

When using LMDz in your configuration the maximum possible PeriodLength is 1M

If SpaceName=TEST all output will be store on scratchdir (on curie) or workdir (on ada)

2.1.2. The section [Executable]

This section contains one line for each model component giving the executable's name in the bin/ directory and the executable's name copied to the working directory. You should only change this section if your executable is running in parallel using MPI and OpenMP or if you have changed the executable's name.

```
[Executable]
#D- For each component, Real name of executable, Name of executable for oasis, optional : number of MPI task, number of Op
ATM= (gcm.e, lmdz.x)
```

```
SRF= ("", "")
...
```

Note : (",") indicates that this component has no executable. It is defined in a library linked to another executable (e.g. Orchidee in LMDZOR or Inca in LMDZINCA).

Example for an MPMD MPI execution with NEMO and XIOS : Ocean on 127 MPI process and IO Server on 1 MPI process.

```
[Executable]
#D- For each component, Real name of executable, Name of executable in RUN_DIR directory, Number of MPI process, Number of
OCE= (opa, opa.xx, 127MPI)
ICE= ("" ,"" )
MBG= ("" ,"" )
IOS= (xios_server.exe, xios.x, 1MPI)
```

Example for an MPMD hybrid MPI/OpenMP execution with IPSLCM coupled configuration : Atmosphere on 27 MPI process and 4 OMP threads per process, Ocean on 19 MPI process, IO Server on 1 MPI process.

```
[Executable]
#D- For each component, Real name of executable, Name of executable in RUN_DIR directory, Number of MPI process, Number of
ATM= (gcm.e, lmdz.x, 27MPI, 4OMP)
SRF= ("" ,"" )
SBG= ("" ,"" )
OCE= (opa, opa.xx, 19MPI)
ICE= ("" ,"" )
MBG= ("" ,"" )
CPL= ("", "" )
IOS= (xios_server.exe, xios.x, 1MPI)
```

Another example for an MPMD hybrid MPI/OpenMP execution with LMDZ and XIOS : Atmosphere on 47 MPI process, 8 OMP threads per processand and IO server on 1 MPI process.

```
[Executable]
#D- For each component, Real name of executable, Name of executable in RUN_DIR directory, Number of MPI process, Number of
ATM= (gcm.e, lmdz.x, 47MPI, 8OMP)
SRF= ("" ,"" )
SBG= ("" ,"" )
IOS= (xios_server.exe, xios.x, 1MPI)
```

Example for an SPMD hybrid MPI/OpenMP simulation with LMDZ : Atmosphere on 32 MPI process and 4 OMP threads per process.

```
[Executable]
#D- For each component, Real name of executable, Name of executable in RUN_DIR directory, Number of MPI process, Number of
ATM= (gcm.e, lmdz.x, 32MPI, 4OMP)
SRF= ("" ,"" )
SBG= ("" ,"" )
```

2.1.3. The [Restarts] section

The Restarts section allow to start from an existing simulation. This simulation can be found at the archive machine or at local scratch- or workdir. Activate by setting **OverRule=y**. All components (e.g. ATM, SRF, etc) will then use the same simulation as restart state.

```
[Restarts]
OverRule=y
RestartDate=1999-12-31                                  # Last day of the experience used as restart for all components
RestartJobName=EXP00                                    # Define restart simulation name for all components
RestartPath=${ARCHIVE}/IGCM_OUT/IPSLCM5A/DEVT/pdControl # Path Server Group Login
```

The root path for the **RestartPath** depend on the computing center. They are:

```
${ARCHIVE}                           # The storage machine of the computing center
                                    # (CCCSTOREDIR or ERGON). This space can contain
                                    # tar of restarts or
                                    # usual restarts files
/ccc/store/cont003/dsm/login         # TGCC
/u/rech/ces/login                    # IDRIS

${SCRATCHDIR}                        # The large TGCC workspace (no backup)

/ccc/scratch/cont003/dsm/login       # This kind of space can contain
                                    # usual restarts files

${WORKDIR}                           # The large IDRIS workspace (no backup)

/workgpfs/rech/ces/login             # This kind of space can contain
                                    # usual restarts files
```

libIGCM manages the difference in treatment between a path pointing to restart files that are directly accessible (without pack) and a path pointing to restart files that are in tar format (after pack). The management is made according to the path you provided.

2.1.4. The [ATM], ..., sections of the model components

This section for each of the model components allows you to:

- **define the output frequency**;
- **define whether this component is installed** which will only be considered if you specified **OverRule=n** in the [Restarts] section.

The possible settings for the **RestartPath** options are the same as for the [Restarts] section.

The possible settings for the **WriteFrequency** options are:

- 1M (monthly)
- 5D (5-day)
- 1D (daily)
- HF (6-hour high frequency)
- HF3h (real-time 3-hour frequency - specific to LMDZ)
- HF3hm (3-hour averaged high frequency - specific to LMDZ)
- STN (instantaneous output only for the CFMIP stations - specific to LMDZ).

```
[ATM]
WriteFrequency="1M 1D"                                  # Activate the writing frequency of this component
Restart=y                                               # If config_Restarts_OverRule == 'n' next 4 params are read
RestartDate=1999-12-31                                  # Last day of the experience used as restart for this component if
RestartJobName=piControl25                              # Define restart simulation name for this component
RestartPath=${ARCHIVE}/IGCM_OUT/IPSLCM5A/PROD/piControl # Path Server Group Login
```

WriteFrequency specific to the model components

- **LMDZ** : ([ATM]) Each of the frequencies settings 1M, 1D, HF, HF3h, HF3hm, and STN correspond to a given output file. For example, if you specify *1M*, a histmth.nc file will be created. If you want to change the output frequency in the histmth file you must change the corresponding lmdz parameter file. See here.

  ORCHIDEE
  - **[SRF]** : The first frequency corresponds to the output frequency for the sechiba_history.nc file. The available frequencies are: xY, xM, 5D, 1D and xs, where x is an integer and s means seconds. This file is required. If you add HF, a second sechiba_out_2.nc file will be written with the 3H frequency.
  - **[SBG]** : Only one frequency (xY, xM, 5D, 1D or xs) can be specified. It applies to the stomate_history.nc file. The stomate_history_ipcc.nc file always contains daily outputs.
- **INCA** : the section WriteFrequency does not work. Click here to learn more about how to change the writing frequency.

2.1.5. The [Post] section

The options of the [Post] section will allow you to set or disable the frequencies for submitting post processing jobs by changing the 5 following options (see the diagram below).

If you do not wish to run post processing jobs, you must specify **NONE** for both **TimeSeriesFrequency** and **SeasonalFrequency**.

RebuildFrequency and **PackFrequency** should not be disabled except in the case of running in expert mode.

```
RebuildFrequency=1Y          # Frequency of rebuild submission (use NONE for DRYRUN=3)
PackFrequency=1Y             # If absent default to RebuildFrequency.
TimeSeriesFrequency=1Y       # Frequency of post-processing submission (NONE if you don't want)
SeasonalFrequency=2Y         # Seasonal average period (NONE if you don't want,
                            # 2Y at least, 10Y by default)
SeasonalFrequencyOffset=0    # Offset for seasonal average first start dates ;
                            # same unit as SeasonalFrequency
```

2.2. COMP directory

This directory contains the architecture (or map) of each model component. Each map specifies inputs and outputs required by a component.

Input files of each component are organized into different **sections**.

- **[UserChoices]** contains specific options. --> used by the component's drivers (e.g.: lmdz.driver)
- **[InitialStateFiles] Initial conditions** files such as vegetation maps, topography,... --> retrieved by the IGCM_comp_GetInputInitialStateFiles function
- **[BoundaryFiles] Boundary conditions** files such as forcings or a LAI --> retrieved by the IGCM_comp_GetInputBoundaryFiles function
- **[SmoothFiles] Time-varying boundary conditions** files such as aerosols --> retrieved by the IGCM_comp_GetInputSmoothFiles function
- **[ParametersFiles] Parameters** files such as namelist or the run.def file --> retrieved by the IGCM_comp_GetInputParametersFiles function
- **[RestartFiles] Restart** files --> retrieved by the IGCM_comp_GetInputRestartFiles function

2.2.1. The [UserChoices] section

Contains several options which change the simulation setup by *drivers* files of the components (lmdz.driver, opa9.driver, ...). For example :

```
[UserChoices]
# Physics package to use :
# LMDZ_Physics=AP for standard/old physics(default), can be used with LMDZ4_AR5 or LMDZ5/trunk sources
# LMDZ_Physics=NPv3.1 for new physics, to be used with LMDZ5/trunk revision 1554 or later
LMDZ_Physics=AP
```

See the description for LMDZ here.

2.2.2. The [InitialStateFiles] section

Files needed to create initial files. This section is not activated if you chose to start or continue from an existing simulation (Section [Restart] in *config.card*). The files in this list will be only copied at the startup of your simulation.

```
# ------------------------------------------------------------------
#D- Get initial state (Etat0, carteveg,relief...)
#D- READ AND USE BY GCM FOR ONLY FOR THE FIRST EXECUTION.
# ------------------------------------------------------------------
[InitialStateFiles]    # IGCM_comp_GetInputInitialStateFiles from main Job
List= (SOURCE, DESTINATION)
```

2.2.3. The [BoundaryFiles] section

The files containing the boundary conditions are copied to the working directory.

The files in the **List** list will be copied at each integration period (one 1-month integration per period in general). A job can consist of several periods (PeriodNb).

The files in the **ListNonDel** list will only be copied for the first period of each job. These files will be accessible but will not change during the simulation.

```
# ------------------------------------------------------------------
#D- Get Boundaries Conditions (SST, WIND[X,Y,Z], LAI ...)
#D- READ AND USE BY GCM AT EACH EXECUTION.
# ------------------------------------------------------------------
[BoundaryFiles]        # IGCM_comp_GetInputBoundaryFiles
List=   (SOURCE, DESTINATION)
ListNonDel= (SOURCE, DESTINATION)
```

Be very careful : if there is any space at the end of a line, libIGCM will not taking in account the next line in the list

2.2.4. The [SmoothFiles] section

These are also files containing boundary conditions but their retrieval is only done at specific time integrals and it is not systematic. **1:12:** means that the file will be copied to the working directory at the first integration step and then every 12 iterations until the simulation is finished.

```
# ------------------------------------------------------------------
#D- Get SmoothFiles Conditions (SST, WIND[X,Y,Z], LAI ...)
#D- READ AND USE BY GCM AT EACH EXECUTION but varying in time
# ------------------------------------------------------------------
[SmoothFiles]          # IGCM_comp_GetInputSmoothFiles
List= (SOURCE, DESTINATION, FREQUENCE DE COPIE)
```

2.2.5. The [ParametersFiles] section

The parameter files of the component (namelist, run.def,...)

```
# ------------------------------------------------------------------
#D- Get parameters text files updated by job (.def, namelist ...)
#D- READ AND USE BY GCM AT EACH EXECUTION.
# ------------------------------------------------------------------
[ParametersFiles]      # IGCM_comp_GetInputParametersFiles
List=   (SOURCE, DESTINATION)
```

2.2.6. The [RestartFiles] section

The files providing the restart data. **You must not change this section** it is needed to link the jobs.

```
# ------------------------------------------------------------------
#D- Get restart files (restartphy.nc, orca_restart.nc ...)
#D- READ AND USE BY GCM AT EACH EXECUTION.
# ------------------------------------------------------------------
[RestartFiles]         # IGCM_comp_GetInputRestartFiles
List=   (MODEL OUTPUT NAME, ARCHIVED NAME, MODEL INPUT NAME)
```

2.2.7. The [OutputText] section

This section contains text files which will be produced during the simulation and model input parameter files. You might want to save these files.

```
[OutputText]
List= (NAME OF TEXT1 FILE, NAME OF TEXT2 FILE ....)
```

This files will be saved in tar stored in the output directory

- TGCC :

$CCCSTOREDIR/IGCM_OUT/TagName/[SpaceName]/[ExperimentName]/JobName/DEBUG

- IDRIS :

$ARCHIVE/IGCM_OUT/TagName/[SpaceName]/[ExperimentName]/JobName/DEBUG

2.2.8. The [OutputFiles] section

The netcdf files produced by the simulation are listed in this paragraph. This paragraph is associated with the [Post_*] **sections.**

```
[OutputFiles]
List = (OUTPUT_FILE_NAME, SAVE_PATH, POSSIBLE ASSOCIATED POST PROCESSING)
```

Refer to this chapter to learn everything about this section.

2.3. DRIVER directory

This directory contains the different *drivers* (predefined libIGCM functions for the component) of the different configuration's components. These drivers modify the parameter files of each component (*.def, namelist, ...) setting the integration times, the outputs, and the forcing files.

Note : If this directory does not exist the *driver* files are located in the COMP directory.

2.4. PARAM directory

This directory contains input text files for the configuration's components.

2.5. POST directory

This directory contains configuration files for additional diagnostic output. Click here for more details.

3. Set up initial state for the simulation

When you setup a simulation make sure that the list of input files in each card file of the model components and the selected options correspond to your experiment.

There are three different ways to define your simulation's initial conditions:

- Start using restart files from an existing simulation by setting *OverRule=y* in the [Restart] section of the config.card file
- Start using different restart files from different simulations for each model component by setting *Restart=y* in each associated part of the *config.card* file. *OverRule=n* must be set in *config.card*.
- Use the default section InitialStateFiles in the comp.card file of the model components in *config.card* you must have *OverRule=n* and and *Restart=n* for this case.

3.1. Example for different restart

3.1.1. Example with OverRule=y

If you wish to use the start state of a given simulation, set in *config.card*:

```
#========================================================================
#D-- Restarts -
[Restarts]
#D- If you want a GENERAL RULE FOR ALL COMPONENTS RESTARTS, put this flag to 'y'
OverRule=y
#D- Last day of the experience used as restart
RestartDate=1869-12-30
#D- Define restart simulation name
RestartJobName=CD1
#D- Path Server Group Login
RestartPath=${ARCHIVE}/IGCM_OUT/IPSLCM5A/DEVT/pdControl
```

For the same case but if the simulation was performed by someone else, you must give the complete path of the directory, for example:

```
RestartPath=/u/rech/lab/plabxxx/IGCM_OUT/IPSLCM5A/DEVT/pdControl
# or RestartPath=/dmnfs/contxxx/login/IGCM_OUT/IPSLCM5A/DEVT/pdControl
```

3.1.2. Example with OverRule=n and [COMP]/Restart=y

You can also distinguish the setup parameters for each model components. Set `OverRule`=n and use the `Restart`, `RestartDate`, `RestartJobName` and `RestartPath` parameters for each model component (section). For example, use restart files for the atmosphere but not for the surface component. For the surface component the InitialStateFiles will then be used :

```
#D-- ATM -
[ATM]
#
WriteFrequency="1M 1D HF"
# If config_Restarts_OverRule == 'n' all params are read
Restart= y
# Last day of the experience used as restart for this component
RestartDate=1999-12-30
# Define restart simulation name
RestartJobName=2L18
RestartPath=${ARCHIVE}/IGCM_OUT/IPSLCM5A/DEVT/pdControl
#
#D-- SRF -
[SRF]
#
WriteFrequency="1M"
# If config_Restarts_OverRule == 'n' all params are read
Restart= n
# Last day of the experience used as restart for this component
RestartDate=1999-12-30
# Define restart simulation name
RestartJobName=2L18
RestartPath=${ARCHIVE}/IGCM_OUT/IPSLCM5A/DEVT/pdControl
```

3.2. Note for LMDZ

To obtain exactly the same outputs in different simulations, you must choose the same LMDZ `Bands` files. This is explained in `COMP/lmdz.card` with the `LMDZ_NbPeriod_adjust` and `LMDZ_Bands_file_name` parameters.

```
LMDZ_NbPeriod_adjust=0
# To force the use of this Bands file, set LMDZ_NbPeriod_adjust=0 and replace XXXXXXX by Restart Job Name
LMDZ_Bands_file_name=${ARCHIVE}/IGCM_OUT/IPSLCM5/CEPRO0/ATM/Debug/CEPRO0_Bands_96x95x39_3prc.dat_3
```

Click here for more details.

4. Main job of the simulation

The main job contains scripts that will be executed by the system. With libIGCM, this job is unique (in the beginning AA_job and later Job_MYJOBNAME) for all type of configurations. It contains all scripts to initialize a simulation, to summarize the chosen model configuration and to run identical experiments for all model components. It resubmits itself in order to continue the simulation if needed.

The job header depends on the machine type. It contains the job name and the parameters. **Real-times** must be chosen to match the specific classes for the computing machine and according to the simulation length (test or production).

At TGCC you must specify the project number: #MSUB -A MY_PROJECT.

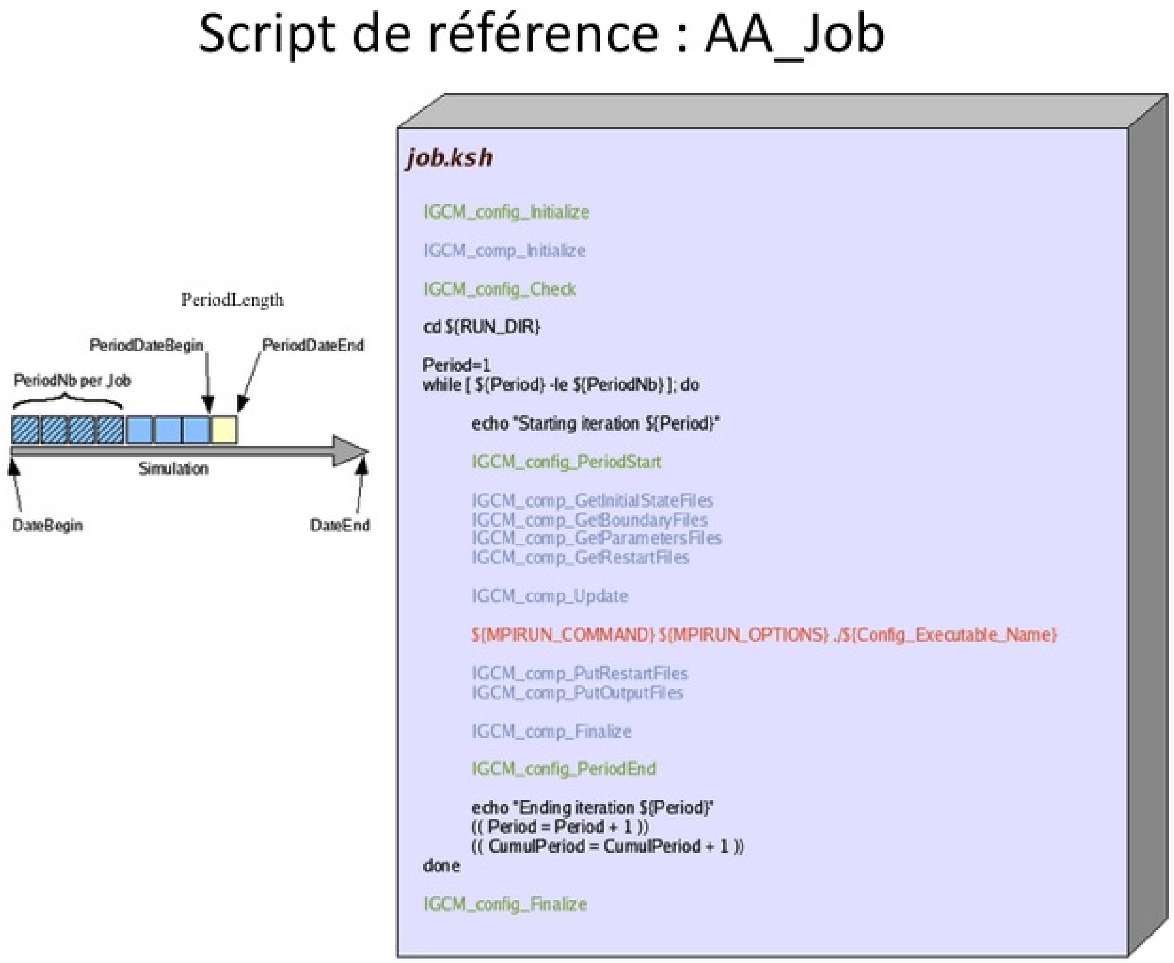
You should change the PeriodNb parameter in the job to change the number of runs in one job (see the example of computation in the next section) :

```
#D- Number of execution in one job
PeriodNb=1
```

In some cases, you must set a variable RUN_DIR_PATH in order to avoid deleting the working directory.

```
#D- Define running directory
#D- Default=${TMPDIR} ie temporary batch directory
#RUN_DIR_PATH=/workdir/or/scratchdir/of/this/machine
```

Here is the diagram of the steps in AA_job :



```
IGCM_config_Initialize

IGCM_comp_Initialize

IGCM_config_Check

cd ${RUN_DIR}

Period=1
while [ ${Period} -le ${PeriodNb} ]; do

        echo "Starting iteration ${Period}"

        IGCM_config_PeriodStart

        IGCM_comp_GetInitialStateFiles
        IGCM_comp_GetBoundaryFiles
        IGCM_comp_GetParametersFiles
        IGCM_comp_GetRestartFiles

        IGCM_comp_Update

        ${MPIRUN_COMMAND} ${MPIRUN_OPTIONS} ./${Config_Executable_Name}

        IGCM_comp_PutRestartFiles
        IGCM_comp_PutOutputFiles

        IGCM_comp_Finalize

        IGCM_config_PeriodEnd

        echo "Ending iteration ${Period}"
        (( Period = Period + 1 ))
        (( CumulPeriod = CumulPeriod + 1 ))
done

IGCM_config_Finalize
```

4.1. Choosing PeriodNb

To avoid starting a lot of short jobs which might be queued, the production job starts n integrations *(PeriodNb)*, whose length are PeriodLength.

These are calculated as followed:
Time limit = PeriodNb * max(Real time of a PeriodLength)

where *Time limit* is the requested time in the job header.

At the end of a simulation, the run.card file returns the used CPU time for each simulation step. This will allow you to perform this computation. It is therefore important, for each simulation with a new configuration, to perform a 1-3 month test to estimate beforehand the CPU time.

5. Prepare a new experiment

There are two ways to prepare a new working directory for your model configuration:

1. Start again from the first step described above by copying the desired *config.card* file to your configuration directory using a new JobName.
2. Copy an existing submission directory, delete the files created by the simulation, and change JobName in *config.card*.

For example:

```
cd modipsl/config/LMDZOR_v5
cp -r DIADEME CHOUCROUTE
cd CHOUCROUTE
rm -f Job_DIADEME run.card Script_Output_DIADEME.000001
vi config.card
  JobName=CHOUCROUTE
../../../util/ins_job # Check and complet job's header
```

The ins_job script allows you to create a submission directory from a *config.card* file or if the directory already exists it allows you to only create the job corresponding to *config.card*. **`ins_job`** will not overwrite a directory or an existing job.

5.1. Post-processing jobs

Jobs headers for post-processing have to be carefully checked, especially elapsed time limits. They are in libIGCM directory (xxx.job) and are adapted for IPSLCM5A with 1Y for RebuildFrequency and PackFrequency. Change time limits if you use larger frequencies.

---

6. Prepare ensembles with ins_job -e

Remark ! What follows applies to IPSLCM5_v5.

When IPSLCM5_v5 is downloaded with ./model IPSLCM5_v5 it will offer the possibility to launch experiments of the decadal type. To prepare an ensemble of simulations copy the config.card and ensemble.card files from the directory:

```
modipsl/config/IPSLCM5_v5/EXPERIMENTS/IPSLCM5/decadal/
```

into the directory:

```
modipsl/config/IPSLCM5_v5
```

Several types of ensemble simulations can be prepared by filling config.card and more importantly ensemble.card. All parameters for ensemble description are in ensemble.card and global simulation template is in the config.card.

6.1. Usage

Check that COMP, POST, PARAM and DRIVER (for IPSLCM5_v5) directories are present in the experiment folder. Once ensemble.card and config.card are correctly filled, to create an ensemble simply type:

> ins_job -e

This will create all the directories of the ensemble and Qsub.xxx.sh, a shell file containing all command to submit all jobs (PeriodNb=60 for all simulations). The Qclean.month/year.xxx.sh are bash files to use clean_month or clean_year script for all simulations.

NOTE: If a directory exists, ins_job won't modify it. If only some directories of the ensemble are presents, it will create the missing ones and complete Qsub.xxx.sh shell file.

6.2. Config.card

The file *config.card* is filled as a regular *config.card* (**`ins_job`** without the -e option).

The important lines for the ensemble set up are in the [UserChoices] section. Make sure that JobName and ExperimentName are filled with proper values.

The variable CalendarType should be consistent with the simulations from which you are initialising or else the restart file with the proper date may not exist.
The variables DateBegin and DateEnd will be overiden by variables present in ensemble.card.

```
#D-- UserChoices -
[UserChoices]
#============================
JobName=v3h4testB
#----- Short Name of Experiment
ExperimentName=v3h4testB
#----- DEVT TEST PROD
SpaceName=DEVT
LongName="IPSLCM5A CMIP5 DEVT phase decadal example with limited outputs."
TagName=IPSLCM5A
#D- Choice of experiment in EXPERIEMENTS directory
ExpType=IPSLCM5/decadal
#============================
#-- leap, noleap, 360d
CalendarType=noleap
#-- Experiment dates : Beginning and ending
#-- "YYYY-MM-DD"
DateBegin=2013-01-01
DateEnd=2022-12-31
#============================
#-- 1Y, 1M, 5D, 1D Period Length of one trunk of simulation
PeriodLength=1M
#============================
#-- Total Number of Processors (minimum is 2 for a coupled configuration)
#JobNumProcTot=4
JobNumProcTot=32
```

A section `[Ensemble]` should also be present. It contains the information that we want to prepare an ensemble simulation with variable EnsembleRun set to y and three unset fields to be filled in the *config.card* of each member after **'ins_job -e** has run.

```
[Ensemble]
#D- Ensemble run ? 'y' or 'n'
#D- If 'y', fill in ensemble.card !!
EnsembleRun=y
EnsembleName=
EnsembleDate=
EnsembleType=
```

6.3. Ensemble.card

There are several sections in *ensemble.card*: `[Ens_PARAMETRIC]`, `[Ens_DATE]` and `[Ens_PERTURB]`.

The choice of ensemble types is done by setting the variable `active` to *y* or *n*.

```
[Ens_PERTURB]
# active=y to use this ensemble type
active=y
```

There are 3 types of ensembles :

- Parametric ensemble which is not implemented yet.
- Date restart ensemble which allows to configure several simulations starting from different conditions.
- Perturb ensemble which allows to generate members from an initial condition which is perturbed by different means.

6.4. Configure a Date Restart ensemble

We cover here the section which allows to generate identical simulations excepted the initial restart file. Indeed, the « Date Restart ensemble » was implemented to configure a same simulation using several restart dates, generally chosen for a particular point (ex : randomly, particular climate oscillation phases, volcanic activity…).

In ensemble.card all configuration items of this ensemble are in `[Ens_DATE]` section.

There are 2 types of possible configurations to define restarts dates : a periodic one (give year start / stop and periodicity) or non periodic one (give a list of desired restarts). The second one is recommanded because it allows more options.

In both cases you must fill the following options : `active`, `NAME`, `LENGTH`, `INITFROM` and `INITPATH`.

```
# for using date ensemble, 'n' else.
Active=y

# name of the ensemble (used to create root directory)
NAME= ENSTAMBORA

# default length of the simulation for non periodic and duration for all periodic (in Year or Month)
LENGTH=10Y

# Experiment name to find all restart files (and default one for non-periodic)
INITFROM=v3.historical6

# Restart root directory
INITPATH=/ccc/store/cont003/dsm/p86denv/dmf_import/IGCM_OUT/IPSLCM5A/PROD/historical
```

Periodic start dates

In ensemble.card, it is possible to specify a periodic list of start dates. Restart files will be generated for each member at each date starting from `BEGIN_INIT` to `END_INIT` with a periodicity of `PERIODICITY`, using `BEGIN_RESTART` as first restart. Leave all *NON_PERIODIC* options empty (`NONPERIODIC`, `RESTART_NONPERIODIC`, `INITFROM_NONPERIODIC`, `LENGTH_NONPERIODIC`).

The following part of ensemble.card sets simulations from 1990-01-01 to 2000-01-01 every 2 years each lasting 10 years with a restarts starting from 1814-12-31 every 2 years:

```
# start date of the first periodic simulation
BEGIN_INIT=19900101

# start date of the last periodic simulation
END_INIT=20000101

# duration between the start of 2 periodic simulations
PERIODICITY=2Y

# date for the first restart (next = first+periodicity). CAUTION of the calendar (use config.card one)!
BEGIN_RESTART=18141231
```

This will produce simulations starting at the dates : 1990-01-01, 1992-01-01, 1994-01-01, 1996-01-01, 1998-01-01, 2000-01-01. (`PERIODICITY` can be given in months for shorter periods).

The restart files are taken from `BEGIN_RESTART` every `PERIODICITY` step : 1814-12-31, 1816-12-31, 1818-12-31, etc...

The directory in which the start date is retrieved is given by `INITPATH` and `INITFROM`.

To restart from experiment *v3.historical6* in directory */ccc/store/cont003/dsm/p86denv/dmf_import/IGCM_OUT/IPSLCM5A/PROD/historical* fill:

```
# Restart name
INITFROM= v3.historical6

# Restart directory
/ccc/store/cont003/dsm/p86denv/dmf_import/IGCM_OUT/IPSLCM5A/PROD/historical
```

Non-Periodic start dates

In ensemble.card, it is also possible to specify manually all simulations running and restart dates, length and experiment names to get restart files.

First, you don’t need to fill the periodic attributes in ensemble.card (`BEGIN_INIT`, `END_INIT`, `PERIODICITY` and `BEGIN_RESTART`). Then you can list the start date of all simulations with `NONPERIODIC` variable, all restart dates with `RESTART_NONPERIODIC` one, all experiments to get restart files in `INITFROM_NONPERIODIC` and give the length of each simulation (`LENGTH_NONPERIODIC`).

Here is an example of a configuration :

```
# list of start dates for all simulations
NONPERIODIC=(18150101 19910101 19990101)

# list of corresponding restart dates
RESTART_NONPERIODIC=(18141230 19901230 19981231)

# simulation name to restart for each simulation. IF empty all simulations will use INITFROM one.
INITFROM_NONPERIODIC=( v3.historical6 v3.historical6 v5.historical1)

# lenght of each simulation. IF empty all simulations duration will be the default LENGTH option.
LENGTH_NONPERIODIC=(10Y 10Y 50Y)
```

WARNING: For list variables, use blank between values if there are not dates or durations (no coma).

This will produce 3 simulations which starting at the dates : 1815-01-01, 1990-01-01 and 1999-01-01using respectively restarts from 1814-12-30, 1990-12-30, 1998-12-31 (note that the calendar should be different from the config.card one) taking into *v3.historical6* experiment for the 2 firsts and from *v5.historical1* for the last one (`INITPATH` is ignored when `INITFROM_NONPERIODIC` is filled). Simulations length will be 10 years for the 2 firsts and 50 years for the last one. All restart experiments should be in the directory */ccc/store/cont003/dsm/p86denv/dmf_import/IGCM_OUT/IPSLCM5A/PROD/historical.*

Notice that `INITFROM_NONPERIODIC` and `LENGTH_NONPERIODIC` are not mandatory for non-periodic configuration. If you don’t fill one of them or both the `INITFROM` value and/or `LENGTH` value will be used for all simulations :

```
# default length of the simulation for non periodic and duration for all periodic (in Year or Month)
LENGTH=10Y
[…]

# list of start dates for all simulations
NONPERIODIC=(18150101 19910101 19990101)

# list of corresponding restart dates
RESTART_NONPERIODIC=(18141230 19901230 19901231)

# simulation name to restart for each simulation. IF empty all simulations will use INITFROM one.
INITFROM_NONPERIODIC=

# length of each simulation. IF left empty all simulations durations will be the default LENGTH option.
LENGTH_NONPERIODIC=

# Restart name
INITFROM= v3.historical6

# Restart directory
/ccc/store/cont003/dsm/p86denv/dmf_import/IGCM_OUT/IPSLCM5A/PROD/historical
```

This will produce 3 simulations starting at the dates : 1815-01-01, 1990-01-01 and 1999-01-01 using respectively restarts from 1814-12-30, 1990-12-30, 1990-12-31.
All of them use *v3.historical6* experiment to get restart files and their duration is 10 years.

6.5. Configure a Perturbed ensemble

We cover here the section which allows to generate members from an initial condition which is perturbed by different means.

There are two ways to perturb the initial condition:

- apply some random white noise of defined amplitude to the temperature field of the coupler component (CPL) restart file
- apply some previously generated 3D temperature perturbation map to the temperature field of the ocean component (OCE) restart file

Each method apply only to the relevant type of ensemble generation available inside [Ens_PERTURB] as will be explained later.

Before detailing the different functionalities available in [Ens_PERTURB] let us discuss the NAME variable.
This variable should be set identical to the JobName variable, otherwise the script will fail to generate the proper files.

```
# ensemble name (must be equal to JobName in config.card)
NAME=v3h4testB
```

Periodic start dates

For this type of perturbed ensembles the following variables are left empty:

```
# member list (apply list of pattern to initial state)
MEMBER_LIST=()

# member list of names corresponding to each member
MEMBER_NAMESLIST=()

# member pattern global name
MEMBER_INITFROM=

# member pattern global directory for name
MEMBER_INITPATH=
...
# start dates list
NONPERIODIC=()
# length list for non periodic simulation (NOTE: use length above if not fill)
LENGTH_NONPERIODIC=()
...
# Path of Mask file
MASKPATH=
```

In ensemble.card, it is possible to specify a periodic list of start dates.
Restart files will be generated for each member at each date starting from BEGIN_INIT to END_INIT with a periodicity of PERIODICITY.
The variable MEMBER sets the number of members for each start date.

The following part of ensemble.card sets 10 members from 19900101 to 20000101 every 2 years each lasting 10 years:

```
# member nb (i.e nb of perturb initial restart for each date)
MEMBER=10
...
# periodic and member list simulations length
LENGTH=10Y
# start date of the first ensemble
BEGIN_INIT=19900101
# start date of the last ensemble
END_INIT=20000101
# timestep between each periodic simulation
PERIODICITY=2Y
```

This will produce 10 members starting at the dates : 19900101, 19920101, 19940101, 19960101, 19980101, 20000101. (PERIODICITY can be given in months for shorter periods)

Each time the restart file to be perturbed in order to produce each member is taken from the previous day of the start date : 19893112, 19913112, etc...

The directory in which the start date is retrieved is given by INITPATH and INITFROM.
To restart from experiment v3h4BTxx in directory /ccc/store/cont003/gen2211/nguyens/IGCM_OUT/IPSLCM5A/PROD/historical fill:

```
# Restart name
INITFROM=v3h4BTxx
# Restart directory
INITPATH=/ccc/store/cont003/gen2211/nguyens/IGCM_OUT/IPSLCM5A/PROD/historical
```

The way the perturbed member is generated depends on PERTURB_BIN array. The first two elements are the most important. The first one is the executable to be used to produce the members, the second one is the component from which the restart is perturbed.

In the Periodic Case it is only possible to build the members by applying a randomly generated temperature pattern on the restart file of the coupler. PERTURB_BIN should look like this:

```
PERTURB_BIN=(AddNoise, CPL, sstoc, O_SSTSST, 0.1)
```

The list is interpreted as follows:

- the used executable is *AddNoise*,
- the component is the coupler (*CPL*),
- the restart file to perturb contains *sstoc* in its name,
- the variable to perturb in the restart file is *O_SSTSST*,
- the randomly generated perturbation is in [-.05;+0.05] degrees

For each member (in our example we have ten) a new restart file for the coupler will be generated using the executable *addnoise* to add some randomly generated temperature perturbation.
For the year 1990, the corresponding restart file of member 1 will be stored in

```
$WORKDIR/IGCM_IN/v3h4testB190/v3h4testB190A/CPL/Restart/
```

Non-Periodic start dates

For this type of perturbed ensembles the following variables are left empty:

```
# member list (apply list of pattern to initial state)
MEMBER_LIST=()

# member list of names corresponding to each member
MEMBER_NAMESLIST=()

# member pattern global name
MEMBER_INITFROM=

# member pattern global directory for name
MEMBER_INITPATH=
...
# start dates list
NONPERIODIC=()
# length list for non periodic simulation (NOTE: use length above if not fill)
LENGTH_NONPERIODIC=()
...
# start date of the first ensemble
BEGIN_INIT=
# start date of the last ensemble
END_INIT=
...
# Path of Mask file
MASKPATH=
```

The variable LENGTH must be set to something but is not used, PERIODICITY must be set to NONE:

```
# periodic and member list simulations length
LENGTH=10Y
...
# timestep between each periodic simulation (NONE for nonperiodic)
PERIODICITY=NONE
```

To set 10 members for the starting dates 1990 and 1992 for a duration of 10 years, set MEMBER, NONPERIODIC and LENGTH_NONPERIODIC as follows:

```
# member nb (i.e nb of perturb initial restart for each date)
MEMBER=10
...
# start dates list
NONPERIODIC=(19900101 19920101)
# length list for non periodic simulation (NOTE: use length above if not fill)
LENGTH_NONPERIODIC=(10Y 10Y)
```

This results in 20 simulations in total.

The restart files to be perturbed to produce each member are sought in directory INITFROM which PATH is INITPATH.

```
# Restart name
INITFROM=v3h4BT00
# Restart directory
INITPATH=/ccc/store/cont003/gen0826/labetoul/dmf_import/IGCM_OUT/IPSLCM5A/PROD/historical
```

This will result in using restarts from experiment *v3h4BT00* located in directory */ccc/store/cont003/gen0826/labetoul/dmf_import/IGCM_OUT/IPSLCM5A/PROD/historical*.

The perturbation executable must be *AddNoise*.

```
PERTURB_BIN=(AddNoise, CPL, sstoc, O_SSTSST, 0.1)
```

List of members for a single start date

For this type of perturbed ensembles the following variables are left empty:

```
# member nb (i.e nb of perturb initial restart for each date)
MEMBER=
# timestep between each periodic simulation (NONE for nonperiodic)
PERIODICITY=NONE
# start dates list
NONPERIODIC=()
# length list for non periodic simulation (NOTE: use length above if not fill)
LENGTH_NONPERIODIC=()
```

It is important to leave PERIODICTY set to NONE and LENGTH_NONPERIODIC as an empty list: the list of member method only works for a single start date and neither with periodic start dates nor with non periodic start dates.

The variables BEGIN_INIT and END_INIT are set to the same date, only BEGIN_INIT will be used to provide the start date of the simulation for each member.

```
# start date of the first ensemble
BEGIN_INIT=20560101
# start date of the last ensemble
END_INIT=20560101
```

The variable LENGTH is the computation time which is the same for all members.

```
# periodic and member list simulations length
LENGTH=10Y
```

MEMBER_NAMESLIST is the list of names given to each member. It gives the names of the subdirectories from which the Job is submitted for each member as well as the subdirectories in which the results are stored for each member.

MEMBER_LIST is the list of perturbation maps files names prefix to apply to the restart file. It is implied that the files are named *prefix.nc*.

MEMBER_INITFROM is the directory in which the perturbations maps are stored.

MEMBER_INITPATH is the path to this directory.

```
# member list (apply list of pattern to initial state)
MEMBER_LIST=(OWN3DT_A, OWN3DT_B, OWN3DT_C, OWN3DT_D)

# member list of names corresponding to each member
MEMBER_NAMESLIST=(OWN3DTA, OWN3DTB, OWN3DTC, OWN3DTD)

# member pattern global directory name
MEMBER_INITFROM=OWN3DTpf

# member pattern global directory for name
MEMBER_INITPATH=/ccc/work/cont003/gen2211/nguyens/PERTU/VECTORS
```

The variables INITFROM and INITPATH are still used to point to the directory where the restart files including the one to be perturbed are available.

```
# Restart name
INITFROM=piControl2

# Restart directory
INITPATH=/ccc/store/cont003/dsm/p86caub/dmf_import/IGCM_OUT/IPSLCM5A/PROD/piControl
```

For the member list perturbation type we use the executable AddPertu3DOCE and set PERTURB_BIN this way:

```
# perturbation type
PERTURB_BIN=(AddPertu3DOCE, OCE, restart, tn, ORCA2_mesh_mask.nc)
```

The elements of the list mean:

- the executable to be called to generate the perturbation is *AddPertu3DOCE*
- the component is the Ocean (*OCE*)
- the restart file to perturb is **restart*.nc*
- the field to perturb in the restart file is *tn*
- the meshmask file to tell if the gridcell is land or sea is *ORCA2_mesh_mask.nc*

The path to the mesh mask file is given in MASKPATH.

```
# Path of Mask file
MASKPATH=/ccc/cont003/home/gen2211/nguyens/addpertu
```

Once config.card and ensemble.card properly filled the directories containing the jobs to launch the simulations are created by issuing the command:

```
ins_job -e # Check and complet job's header
```