

## Ensemble setup

---

This chapter describes how to setup your simulation once you have compiled your configuration at a chosen resolution.

### Table of Content

<b>Ensemble setup</b>	<b>1</b>
<b>1. Prepare ensembles with ins_job -e</b>	<b>2</b>
1.1. Usage	2
1.2. Config.card	2
1.3. Ensemble.card	3
1.4. Configure a Date Restart ensemble	4
1.5. Configure a Perturbed ensemble	4
Periodic start dates	5

In this chapter we describe how to generate/use **ensemble** simulation.

## 1. Prepare ensembles with `ins_job -e`

To create an ensemble configuration you need to create an `ensemble.card` file.

**NOTE:** a template of `ensemble.card` is given in `IPSLCM6/EXPERIMENTS/IPSLCM/dcppAhindcast_CMIP6` for `IPSLCM6` model.

When `IPSLCM6.1.9-LR` is downloaded with `./model IPSLCM6.1.9-LR` 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:

```
modips1/config/IPSLCM6/EXPERIMENTS/IPSLCM/dcppAhindcast_CMIP6
```

into the directory:

```
modips1/config/IPSLCM6/
```

Two type of esemble are allowed with this template:

1. `Ens_DATE` : allows to configure simulations starting from different restart dates, `NON-Periodic`.
1. `Ens_PERTURB` : allows to configure simulations of differents members from an initial condition which is perturbed, `PERIODIC`, (it is possible to create different initial state perturbing Sea Surface Temperature with a with noise)

All parameters for ensemble description are in [ensemble.card](#) and in [config.card](#).

### 1.1. Usage

Check that `COMP`, `POST`, `PARAM` and `DRIVER` directories are present in the experiment folder.

Once [ensemble.card](#) and [config.card](#) are correctly filled, to create an ensemble simply type:

```
../libIGCM/ins_job -e
```

This will create :

- all the directories of the ensemble
- `Qsub.xxx.sh` : shell script to submit all jobs (`PeriodNb=5` for all simulations)
- `Qclean.PeriodLengt.xxx.sh` : shell script to clean (if necessary) all simulations after an error, and to re-launch them
- `Qclean.latestPackperiod.xxx.sh` : shell script to clean (if necessary) last packed period after an error, and to re-launch them

**Note:** `xxx` it will be the `JobName` configured in `config.card`

NOTE: If a directory exists, `ins_job` won't modify it, the creation of ensemble it will stopped.

### 1.2. Config.card

The file `config.card` is filled as a regular `config.card` (`ins_job` without the `-e` option). It will be used as a template for all simulations that will be created.

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 variables `DateBegin` and `DateEnd` will be overridden by variables present in [ensemble.card](#).

```
#
# This is config.card file for IPSLCM6 configuration
#
#=====
#D-- Compatibility -
[Compatibility]
libIGCM=1.0
#D-- UserChoices -
[UserChoices]
```

```

#=====
JobName=CM619-LR-dcppA-hindcast

#----- Short Name of Experiment
ExperimentName=dcppA-hindcast

#----- DEVT TEST PROD
SpaceName=DEVT

LongName="IPSLCM6.1.9-LR"

TagName=IPSLCM6

ModelName=IPSL-CM6A-LR

Member=r1ilplf1

#D- Choice of experiment in EXPERIMENTS directory
ExpType=IPSLCM/dcppAhindcast_CMIP6
#=====

#-- leap, noleap, 360d
CalendarType=leap

#-- Experiment dates : Beginning and ending
#-- "YYYY-MM-DD"
DateBegin=1961-01-01
DateEnd=1980-12-31

#=====
ORCA_version=eORCA1.2

#=====
#-- 1Y, 1M, 5D, 1D Period Length of one trunk of simulation
PeriodLength=1Y

```

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=
#

```

### 1.3. Ensemble.card

It is possible to choose between 2 types of ensembles in *ensemble.card*: Date restart ensemble which allows to configure simulations starting from different restart dates: need to fill section [Ens\_DATE] Perturb ensemble which allows to generate members from an initial condition which is perturbed by different means: need to fill section [Ens\_PERTURB]

The choice between 2 types of ensembles set the variable active to y of section that you want to activate:

To activate Date Restart (non-Periodic) Ensembles:

```
[Ens_DATE]
active=y

[Ens_PERTURB]
active=n
```

To activate Perturbed Restart (Periodic) Ensembles:

**NOTE :** that Period can be also just 1Y (to run non-periodic perturbed ensembles)

```
[Ens_DATE]
active=n

[Ens_PERTURB]
active=y
```

## 1.4. Configure a Date Restart ensemble

The « Date Restart ensemble » is implemented to configure a set of simulations 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.

The configuration to define restarts dates is non-periodic one, it needs to define a list of desired restarts.

Fill the following options : active, NAME, LENGTH, INITFROM and INITPATH.

```
[Ens_DATE]
# active=y to use date ensemble, 'n' for no DO NOT use.
active=y

#--- Default values for all members ---
# name of the ensemble (used to create root directory)
NAME= CM618-LR-volc-pinatubo-full

# default length for all simulations
LENGTH=10Y

# Default start date for all simulations
STARTDATE=19900601

# Experiment name to find all restart files (and default one for non-periodic)
INITFROM=CM61-LR-pi-03

# Restart root directory
INITPATH=$CCCHOME/IGCM_OUT/IPSLCM6/REDO/piControl/CM61-LR-pi-03-REDO.MAY/1870-01-01

#--- Specific values for each member (overrule default) ---
# list of corresponding restart dates
RESTART_NONPERIODIC=(18700531 18810531)
```

**CAUTION:** The variable CalendarType from *config.card* will be used to determine the next restart date. It should be consistent with the simulations from which you are initialising.

**WARNING:** For list variables, use blank between values (no coma).

## 1.5. Configure a Perturbed ensemble

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

```
# ensemble name (must be equal to JobName in config.card)
NAME=CM619-LR-dcppA-hindcast

# member nb (i.e nb of perturb initial restart for each date)
MEMBER=10

MEMBER_LIST=( )

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

# member pattern global directory name
MEMBER_INITFROM=

# member pattern global directory for name
MEMBER_INITPATH=

# periodic and member list simulations length
LENGTH=10Y

# start date of the first ensemble
BEGIN_INIT=19610101

# start date of the last ensemble
END_INIT=19801231

# timestep between each periodic simulation
PERIODICITY=2Y
```

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 applies 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 will be both the global name of the ensemble (ie directory name) and the prefix for each member:

```
# ensemble name
NAME=v3h4testB
```

JobName variable in config.card will be the name of the root directory that would be created containing all config and script files and the ensemble.

### Periodic start dates

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

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

**!!NOTA!! The perturbation is not applied to grid points located under the sea ice. This condition is "hard-written" in the AddNoise code. Because of a change of the name of the sea ice cover variable from IPSL-CM5A (OlceFrac) and IPSL-CM6 (OlceFrc), a modification of the code has been made by Olivier Marti in June 2016 to allow the code to search for both names**

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/
```

## ADD something on SST perturbed can be found in IGCM\_OUT/CPL/.....

The perturbation executable must be *AddNoise*.

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

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
```