

# Compile

---

The compiling steps are described in this chapter.

## Table of Content

<b>Compile</b>	<b>1</b>
<b>1. Compiling environment</b>	<b>2</b>
1.1. FCM	2
<b>2. Makefile for configurations _v6 and _v6.1.*</b>	<b>2</b>
2.1. Creating the main makefile	2
2.2. The main makefile	3
2.3. How to compile?	4
2.4. How to change compile options for debugging or optimization of the model?	4
<b>3. Script for configurations _v6.2 and newer</b>	<b>5</b>
3.1. Compile using a main script	5
3.2. Link to the execution	6

## 1. Compiling environment

First of all, make sure that your working environment contains all the necessary compiling tools. To this end, see the chapter [Computing environments](#)

### 1.1. FCM

Most of IPSL models and components use the **fc**m tool for compiling. Recent versions of the models have fcm in their source directory. Therefore no specific installation is needed anymore.

If no fcm is available in the source directory, fcm must then be installed and added to your environment variable PATH. At following machines, fcm is already installed

```
# At TGCC:
PATH=`ccc_home -u igcmg`/Tools/fcm/bin:$PATH

# At IDRIS:
PATH=

# At LSCE(obelix):
PATH=/home/orchideeshare/igcmg/fcm/FCM_V1.2/bin:$PATH
```

For other machines, extract fcm from svn and add the path to your environment variable PATH. Extract as follow:

```
svn co http://forge.ipsl.jussieu.fr/fcm/svn/PATCHED/FCM_V1.2 fcm
```

For each machine supported by the IPSL tools, 3 kinds of files **\*.fcm**, **\*.path**, and **.env** exist. They provide all the information specific to a machine needed by fcm to create a makefile. These files are stored in the directory arch/ (LMDZ/arch/, INCA/arch/).

- \*.fcm : contains the compiler information
  - compiler name
  - compiling options
  - options and settings
- \*.path : contains the library path names.
- \*.env : contains module to be load to install properly the environment for the compilation of this model.

These files have the name of the machine as a prefix : arch-X64\_IRENE.fcm, arch-X64\_IRENE.path

**Note** : for configurations IPSLCM, arch files for OASIS and NEMO are taken in modipsl/config/IPSLCM\*/SOURCE/OASIS and modipsl/config/IPSLCM6/SOURCE/NEMO

## 2. Makefile for configurations \_v6 and \_v6.1.\*

### 2.1. Creating the main makefile

Each model configuration XXX has a main makefile which provides information about each model component's makefiles. The main Makefile is stored in the modipsl/config/XXXX directory. The main Makefile is created when a model configuration is downloaded and installed.

Example :

```
cd modipsl/util
./model LMDZOR_v6.1.10
(...)
Makefiles setup, scripts and data for irene

Installation in ../config/LMDZOR_v6
Path from Makefile to modipsl/util : ../../util
```

You can recreate those Makefiles by running the `ins_make` script. You have to do this if, for example, you changed the path of `modipsl`.

```
cd modipsl/util/
./ins_make
```

The main Makefile is created by concatenating the following 3 items: `AA_make.ideal` and `AA_make` located in the directory where the main Makefile will be created, and the lines corresponding explicitly to the target machine of the `util/AA_make.gdef` file. For each predefined machine the `util/AA_make.gdef` file contains the compiler settings. If your machine is not included, you can choose a target among the predefined machines in `AA_make.gdef` or you can add a new one.

```
vi AA_make.gdef

(...)
#-Q- jeanzay
(...)
#-Q- irene
(...)
#-Q- aix6
(...)
#-Q- gfortran
(...)
```

In this case, the script becomes

```
./ins_make -t target
```

## 2.2. The main makefile

The Makefile is available for different resolutions. Example with `LMDZOR_v6.1.10` :

```
LMD5655 : libioipsl xios liborchidee lmdz56x55x19 verif
echo "noORCAxLMD5655" >.resol
echo "RESOL_ATM_3D=56x55x19" >>.resol

LMD9671 : libioipsl xios liborchidee lmdz96x71x19 verif
echo "noORCAxLMD9671" >.resol
echo "RESOL_ATM_3D=96x71x19" >>.resol

LMD9695 : libioipsl xios liborchidee lmdz96x95x19 verif
echo "noORCAxLMD9695" >.resol
echo "RESOL_ATM_3D=96x95x19" >>.resol

LMD9695-L39 : libioipsl xios liborchidee lmdz96x95x39 verif
echo "noORCAxLMD9695-L39" >.resol
echo "RESOL_ATM_3D=96x95x39" >>.resol

LMD9695-L79 : libioipsl xios liborchidee lmdz96x95x79 verif
echo "noORCAxLMD9695-L79" >.resol
echo "RESOL_ATM_3D=96x95x79" >>.resol

LMD128118-L39 : libioipsl xios liborchidee lmdz128x118x39 verif
echo "noORCAxLMD128118-L39" >.resol
echo "RESOL_ATM_3D=128x118x39" >>.resol

LMD144142 : libioipsl xios liborchidee lmdz144x142x19 verif
echo "noORCAxLMD144142" >.resol
echo "RESOL_ATM_3D=144x142x19" >>.resol

LMD144142-L39 : libioipsl xios liborchidee lmdz144x142x39 verif
echo "noORCAxLMD144142-L39" >.resol
```

```

echo "RESOL_ATM_3D=144x142x39" >>.resol

LMD144142-L59 : libioipsl xios liborchidee lmdz144x142x59 verif
echo "noORCAxLMD144142-L59" >.resol
echo "RESOL_ATM_3D=144x142x59" >>.resol

LMD144142-L79 : libioipsl xios liborchidee lmdz144x142x79 verif
echo "noORCAxLMD144142-L79" >.resol
echo "RESOL_ATM_3D=144x142x79" >>.resol

LMD512360-L79 : libioipsl xios liborchidee lmdz512x360x79 verif
echo "noORCAxLMD512360-L79" >.resol
echo "RESOL_ATM_3D=512x360x79" >>.resol

```

Available resolutions are :

- LMD5655
- LMD9671
- LMD9695
- LMD9695-L39
- LMD9695-L79
- LMD128118-L39
- LMD144142
- LMD144142-L39
- LMD144142-L59
- LMD144142-L79
- LMD512360-L79

### 2.3. How to compile?

Once you chose a resolution you can start compiling:

```
gmake my_resolution
```

Example for LMDZOR\_v6.1.10:

```
cd modipsl/config/LMDZOR_v6
gmake LMD144142-L79
```

An hidden file `.resol` is created when compiling was successful. This file contains information about the resolution you have just compiled. This file will be used later when setting up the simulation, in particular to locate the input files.

Going back to the previous example, the `.resol` file looks as follows :

```
noORCAxLMD144142-L79
RESOL_ATM_3D=144x142x79
```

Once the file `.resol` is created, you can recompile your configuration using the gmake script without specifying the resolution.

For ORCHIDEE\_OL, the resolution is not specified during run time. Use only `gmake` to compile. The file `.resol` will not be created.

### 2.4. How to change compile options for debugging or optimization of the model?

- You can change your optimization options for IOIPSL models in the `AA_make.gdef` file. To this end you must find your target machine and change the associated lines. You will have to recreate the Makefile.
- Three levels of optimization for the ORCHIDEE, INCA and LMDZ models are predefined in the `arch/files`. You need to write them in the `modipsl/config/.../Makefile` file. The three options are: **-debug**, **-dev**, **-prod**. XIOS offers the same three levels of compilations, accessible through the flags `--debug`, `--dev` and `prod`.

Example: to compile LMDZ you can add the keyword "-debug" :

```
(cd ../../modeles/LMDZ; ./make1mdz_fcm (...) -debug (...) -arch ${FCM_ARCH} gcm ; cp bin/gcm_${RESOL_LMDZ}_phylmd_para_orc
```

The default option is `-prod`

### 3. Script for configurations `_v6.2` and newer

#### 3.1. Compile using a main script

The compilation differs from what has been done so far. Now we use a script `compile_ConfigName.sh` with several options ( resolution, level of optimization, full or partial recompilation ) to compile the model.

- IPSLCM6.2 : `compile_ipslcm6.sh`
- LMDZOR\_v6.2 and LMDZOR\_v6.3 : `compile_lmdzor.sh`
- LMDZORINCA\_v6.2 : `compile_lmdzorinca.sh`
- ICOLMDZOR\_v7 : `compile_icolmdzor.sh`
- ORCHIDEE\_OL using the trunk ORCHIDEE from revision 6526 and ORCHIDEE\_3 : `compile_orchidee_ol.sh`

For each script you can use the option "-h" to know the usage of the script. You can find a description of each script on its configuration page (for example [IPSLCM6.2](#))

See here some examples for compilation:

```
cd modipsl/config/ConfigName

Default compilation, default resolution and in production mode will be set:
./compile_ConfigName.sh

Compilation in debug mode, default resolution:
./compile_ConfigName.sh -debug

Complete recompilation:
./compile_ConfigName.sh -full
```

For machines where the configuration has been adapted, the configuration is not longer depending on the users environment, there is no need to ensure loading the correct modules before compilation. Instead, in the beginning of the compilation script, the environment will be loaded using a specific architecture dependent file in ARCH directory.

```
> ls modipsl/config/IPSLCM6
ARCH/
compile_ipslcm6.sh
EXPERIMENTS/
GENERAL/
SOURCES/
> ls modipsl/config/IPSLCM6/ARCH/
arch-X64_IRENE.env
arch-X64_JEANZAY.env
```

The compilation script finds out which is the file to be used from ARCH folder and sets a link from `arch-${FCM_ARCH}.env` to `arch.env`. This is done only on machines where the configuration has been adapted.

For a local machine, it is possible to add a file ARCH/arch-myarch.env and compile with argument:

```
./compile_ConfigName.sh -arch myarch
```

At the end of the compilation the binaries will be store with a specific name : `binary_${ResolAtm}_${OptMode}`, where `ResolAtm` is the resolution of LMDZ model, and `OptMode` the level of optimization (prod, debug, dev) use during the compilation.

```
> ls modips1/bin
gcm_144x142x79_prod.e
xios_server_prod.exe
opa_ORCA1_prod
```

### 3.2. Link to the execution

It is important to use the same modules during compilation and run time. The file **arch.env** linked by the compilation script is also set in config.card and will be used by libGCM to ensure the same running environment. In the beginning of config.card, following is set:

```
#-- Source following file with module settings, only if it exists
EnvFile=${SUBMIT_DIR}/../ARCH/arch.env
```

If there is no ARCH/arch.env file, libGCM will source a default file which corresponds to the one used for v6.1 configurations.

Further down in config.card, following variables used to copy the right executables are specified, here for the IPSLCM6 configuration:

```
#-- ResolAtm and ResolOce indicates the atmospheric and ocean resolution
#   These variables are used in the executable name
ResolAtm=144x142x79
ResolOce=ORCA1
=====
# OptMode indicates the optimization mode chosen during compilation
# This variable is used in the executable name
OptMode=prod
```

They are used to copy the executables. If you compile in default mode, no changes are needed. But if you change the resolution or the optimization mode, change these variables accordingly.

If you compile in debug or in dev mode you need to modify the parameter OptMode in config.card before submitting the job. If you change the resolution at the compilation you need to modify the parameter ResolAtm in config.card