# NEMO\_v3.6安装指南

## Required libraries

To install NEMO v3.6, the following libraries are required:

* MPI
* HDF5
* NETCDF4
* XIOS

To compile NEMO3.6, XIOS IO server should be installed. XIOS needs NETCDF4 and if you want to use the “one\_file” mode which means having one overall output instead of outputs for each processor, you will need the hdf/netcdf libraries properly compiled to allow parallel IO. In this document, sample scripts are provided for installing required libraries using Intel compilers.

## Installing XIOS

Obtain the latest revision of XIOS:

svn co http://forge.ipsl.jussieu.fr/ioserver/svn/XIOS/trunk XIOS

Follow instructions given here to install:

<http://forge.ipsl.jussieu.fr/ioserver/wiki/documentation>

# Compiling NEMO v3.6

## 下载代码

Download the latest version from the trunk repository:

svn --username yourusername co http://forge.ipsl.jussieu.fr/nemo/svn/trunk/NEMOGCM

Your username is the same as the one you use for <http://www.nemo-ocean.eu/> website

## 编译代码（网站提供）

To compile NEMO v3.6 you first need to create an architecture file compatible with your machine which also indicates the path to NETCDF4 and XIOS libraries. You can find example architecture files in NEMOGCM/ARCH folder.

After creating the architecture file, compile and create executable using an existing configuration. For example to use GYRE configuration and create a configuration called MY\_GYRE:

cd NEMOGCM/CONFIG

./makenemo –m your\_architecture –r GYRE -n MY\_GYRE

## 编译路线分析

./makenemo -r ORCA2\_LIM3 -n AGRIF -m gfortran\_linux add\_key 'key\_agrif'

-r 后面是算例名称；输入cpp\_ORCA2\_LIM3.fcm，其中是启用的key\_\*

-n 后面是创建一个配置算例（自己起名字）；

-m 后面是使用的编译架构ARCH，使用的编译器和编译参数以及需要链接使用的第三方库的路径；

add\_key: 就是临时需要启用的key\_\*

# How to extract and compile XIOS

* svn co [​http://forge.ipsl.jussieu.fr/ioserver/svn/XIOS/trunk](http://forge.ipsl.jussieu.fr/ioserver/svn/XIOS/trunk) XIOS ; cd XIOS
* ./make\_xios --help (or ./make\_xios -h)

make\_xios - installs XIOS on your architecture

make\_xios [options]

options :

[--prod] : compilation in production mode (default)

[--dev] : compilation in development mode

[--debug] : compilation in debug mode

--arch arch : to choose target architecture

[--avail] : to know available target architectures

[--full] : to generate dependencies and recompile from scratch

[--use\_oasis] : to use Oasis coupler

[--doc] : to generate Doxygen documentation (not available yet)

[--job ntasks] : to use parallel compilation with ntasks

[--netcdf\_lib 'netcdf4\_par'/'netcdf4\_seq' : default netcdf4\_par] : choice of netcdf library

Known architectures : **./make\_xios --avail**

GCC\_LINUX

GCC\_MACOSX

IA64\_PLATINE

PW6\_VARGAS

X64\_CURIE

X64\_HORUS

X64\_TITANE

X64\_TITANE\_GCC

X64\_TITANE\_VAMPIR

X86\_LOCGNU\_FEDORA

X86\_LOCINTEL\_FEDORA

XT6\_LYNX

* Compilation
  + On an known architecture. For example : **./make\_xios --arch X64\_CURIE**
  + On an new architecture. You have to create your "arch\_NEW\_ARCH.fcm" and "arch\_NEW\_ARCH.path" files
* Choice of compilation mode. As explained in the "help", there are three compilation modes :
  + [--prod] : compilation in production mode (default)
  + [--dev] : compilation in development mode
  + [--debug] : compilation in debug mode

For example : **./make\_xios --dev --arch PW6\_VARGAS**

* Use of "arch\*.env" file. "arch\*.env" file is an optionnal file (used only if it exists) which allows to define a compilation evironment in order to compile XIOS server.

For example : cat arch-X64\_TITANE\_GCC.env

module unload bullxmpi

module load openmpi/1.4.2\_gnu

module unload intel

module load gcc

export HDF5\_INC\_DIR=$WORKDIR/hdf5/include

export HDF5\_LIB\_DIR=$WORKDIR/hdf5\_gnu/lib

export NETCDF\_INC\_DIR=$WORKDIR/netcdf4/include

export NETCDF\_LIB\_DIR=$WORKDIR/netcdf4\_gnu/lib

* Use of netcdf library. It is possible to use netcdf in parallel mode or sequential mode :
  + --netcdf\_lib netcdf4\_seq : only the multiple\_file writing mode is available.
  + --netcdf\_lib netcdf4\_par : both one\_file mode and mutliple\_file mode are available.