

Compilation Instruction for DelPhi on Linux/Mac OSX (10.6 or up)/Windows

If there is any question, please go to:

<http://compbio.clemson.edu/forum/>

or

e-mail to delphi@g.clemson.edu

DelPhi Development Team, Clemson University
June 2018

Before compilation:

Besides the source code, we provide executable files for Linux, Windows and Mac Versions of DelPhi. The source code and executable files can be downloaded from the DelPhi website: <http://compbio.clemson.edu/delphi>

Most of the users can use the executable files without compilation. If the executable files cannot be executed on some users' systems, or some advanced DelPhi users need to modify the source code, please see the DelPhi compilation section below:

DelPhi compilation:

DelPhi C++ version:

- **Linux Version:**

1. Updating gcc (optional):

On Linux systems, please check the GCC version. Users can use command:

```
gcc -v
```

to check the version. If the gcc version is lower than 4.4, please update it.

2. Installing boost C++ library:

Boost C++ library can be installed in different ways on different Linux distributions.

Here we provide one of the most general ways:

Download library from: http://www.boost.org/users/history/version_1_55_0.html

Unpack the [boost_1_55_0.tar.gz](#) and you will see folders such as boost, doc, libs, more, status ... etc under the unpacked boost_\$(ver) folder (e.g., boost_1_55_0/) [where, \$(ver) stands for the version number]

Then, check your local Linux distribution

And copy the “boost” folder to be found under the unpacked boost_\$(ver) folder into **/usr/include** or **/lib/include** or **/usr/local/include**¹

(whatever is appropriate for your particular linux distribution).

¹ This step will require ‘root’ privilege, In most generic linux, the relevant directory should be: **/usr/include**

The boost library also needs to be in your executable path (check this in the `~/bashrc` or `~/cshrc` or similar profiles)

3. Compilation:

When C++ compiler and boost libraries are ready, you can compile the distribution of DelPhi with or without openMP.

If you don't need openMP parallelized DelPhi, go to “**Release**” folder, the type the commands:

```
make clean
make
```

The executable file for DelPhi will be created.

For compiling openMP parallelized DelPhi, first turn on the openMP flag on by editing the file “`./src/interface/environment.h`”.

There is a line

```
//#define PARALLEL_OMP
```

Un-comment the line by deleting the “`///” at the beginning and save the file.`

Go to the folder “**Release_omp**”, then type:

```
make clean
make
```

The executable file for openMP parallelized DelPhi will be generated.

You may then wish to rename and copy the executable(s) to your `~/bin/` and access them globally from your user account.

- **Windows Version:**

1. Installation of gcc:

There are several tools to install gcc compiler on Windows. Here we take MinGW as an example to install gcc compiler and compile DelPhi on Windows.

Go to MinGW's website (<https://sourceforge.net/projects/mingw-w64/>) to download the MinGW and install it on your computer. The default installation directory is “`C:\MinGW`”.

2. Adding “`C:\MinGW\bin`” to the PATH environment variable.

Control Panel → search “environment variables” → click Edit the system environment variable → click “Advanced” → click “Environment Variables” → select “Path” → click “Edit”

“`C:\MinGW\bin`” to the PATH environment variable, use “`;`” to “separate `C:\MinGW\bin`” and other existing paths.

3. Download boost C++ library:

DelPhi compilation needs boost C++ library, which can be downloaded from:

https://www.boost.org/users/history/version_1_55_0.html

After download, unzip the [boost_1_55_0.zip](#) and you will see folders such as boost, doc, libs, more, status...

Copy the “boost” folder to C:\MinGW\include

4. Compilation:

Open a command window (cmd), Go to the "**delphicpp/Release**" folder then type command:

```
mingw32-make.exe
```

The executable file will be created after a few minutes.

- **Mac Version:**

To compile DelPhi C++, you will need:

1. Xcode Command Line Tools
2. Homebrew (Optional)
3. GNU C++ Compiler (4.4 or up) with OpenMP supported.
4. Boost C++ Libraries (1.47 or up).

With XCode 4 or 5 you will need to download the command-line tools as an additional step. You will find the option to download the command-line tools in XCode's Preferences. On 10.9 Maverick you can get the command-line tools by simply typing

```
xcode-select --install
```

If you want to use homebrew to install gcc and boost library, make sure you have home installed. If not, please refer to its website for installation: <http://brew.sh>

To install GNU C++ compiler on OSX, use Mac homebrew to install as shown below:

```
brew install gcc
```

Or alternatively, please refer to the HPC Mac OSX website: <http://hpc.sourceforge.net> and download the preferred version to install.

The GNU C++ compiler will be installed in **/usr/local/bin** folder and you may need to rename the symbolic g++-4.9 to g++ (this step is for homebrew installed gcc).
And export the PATH in your **~/.bash_profile**

```
export PATH=/usr/local/bin:$PATH
```

To install Boost C++ Libraries on OSX, use Mac homebrew as shown below:

```
brew install boost
```

Or, alternatively, please refer to the detailed installation instruction provided on the official boost C++ libraries website: http://www.boost.org/doc/libs/1_47_0/doc/html/quickbook/install.html

The Boost C++ libraries will be installed in **/usr/local/include** folder.

When C++ compiler and boost libraries are ready, you can compile the distribution of DelPhi with or without openMP. If you don't need openMP parallelized DelPhi, go to "**Release**" folder, then type:

```
make clean
make
```

Then press enter key. The executable file of DelPhi will be generated.

For compiling openMP parallelized DelPhi, first turn on the openMP flag on by editing the file `./src/interface/environment.h`. There is a line

```
//#define PARALLEL_OMP
```

Un-comment the line by deleting the `"/"` at the beginning and save the file.

Go to the folder `"Release_omp"`, then type:

```
make clean
make
```

The executable file for openMP parallelized DelPhi will be generated.

DelPhi Fortran version:

Linux Version:

1. Updating gcc (optional):

On Linux systems, please check the GCC version. Users can use command:

```
gcc -v
```

to check the version. If the gcc version is lower than 4.4.0, please update it.

2. Compilation:

Go to the directory which contains `"src"` folder and the `"makefile"` file and then type the command:

```
make clean
make
```

The executable file will be generated after a few minutes.

You may then wish to rename and copy the executable(s) to your `~/bin/` and access them globally from your user account.

Windows Version:

1. Installation of gcc:

There are several tools to install gcc compiler on Windows.

Here we take MinGW as an example

to install gcc compiler and compile DelPhi on Windows.

Go to MinGW's website (<https://sourceforge.net/projects/mingw-w64/>) to download the MinGW and install it on your computer. The default installation directory is `"C:\MinGW"`. Add `"C:\MinGW\bin"` to2.

2. Compilation:

Open a command window, go to the directory which contains `"src"` folder and the `"makefile"`

fe, then type the command:

```
mingw32-make.exe
```

The executable file will be generated after a few minutes.

Mac Version:

Make sure that the gcc and gfortran compilers are installed on your system.

1. Updating gcc and gfortran (optional):

The compilation has been tested on osx 10.6 and 10.7, with gcc/gfortran 4.6 or higher version

Make sure downloading the correct gcc version, users can download the gcc/gfortran from the link: <http://hpc.sourceforge.net/>

2. Compilation

Go to the directory which contains "src" folder and the "makefe" file, then type the command:

```
make
```

The executable file will be generated after a few minutes.

Notes on the New Delphi Release (delphicpp_mpi_v5)

Current DelPhi C++ (**Version 8.0**) allows users to compile the DelPhi program into different versions by turning on and off appropriate flags from a single distribution. After downloading the DelPhi source code from DelPhi website, the following compilers / libraries must be pre-installed before the DelPhi program is compiled:

1. appropriate C++ compiler such as GCC 5.4.0 and above;
2. boost library installed in **/usr/include** and its path is recognized in the user environment;
3. latest version of OpenMPI or MPICH if MPI version of DelPhi executable is desired.

To compile the DelPhi C++ into:

a. a regular executable (simple single thread/CPU version):

1. go to the folder **Release** and run the commands:

```
make clean  
make
```

2. an executable named delphicpp_release will be generated when the compilation process is finished.

b. a multi-threading OpenMP executable (able to utilize the maximum computing power of a multi-core CPU to accelerate the calculations):

1. go to the source code folder **src** and uncomment the line "**//#define PARALLEL_OMP**" in the file of **./src/interface/environment.h**;
2. go to the compilation folder **Release_omp** and run the commands:

```
make clean  
make
```

3. an executable named delphicpp_omp_release will be generated when the compilation process is finished.

c. a multi-CPU MPI executable (able to utilize the computing power of CPUs across multiple computing nodes on one HPC cluster):

1. go to the source code folder **/src** and uncomment the line "**//#define PARALLEL_MPI**" in the file of **./src/interface/environment.h**;
2. Go to the folder **Release_mpi** and run the commands:

```
make clean  
make
```