
authors: J. Van Bever

How to install ABINIT on Ubuntu

This file describes how to install ABINIT on Ubuntu. Other linux based systems are similar, but certainly the typical library structure can differ. Explicit testing was done with Ubuntu 19.10. The options presented here are installation by

- Using apt
- Compiling from source

Other possibilities (such as installation using anaconda) are not discussed here.

Using apt

To install ABINIT, just type:

```
sudo apt install abinit
```

... type your user password and ABINIT should install smoothly with its dependencies.

WARNING!

- This version of ABINIT is most likely an older one.

Compiling from source under Ubuntu

Background (optional section)

The newest release of ABINIT is available at <https://www.abinit.org/packages>. Compilation from source enables one to optimize parallelism, customize linear algebra libraries etc. Any of the prerequired packages are available via apt. Simply type

```
sudo apt install [package]
```

If you want to search a package or look for more details about a package, you can use the commands `apt search` and `apt show` respectively.

Packages can contain for example executable files that are placed in a bin folder (e.g. `/usr/bin`) and their location can be requested via the unix `which` command. For example

```
which gfortran
```

will show the location of the gfortran executable or nothing if the gfortran package was not installed.

Other packages of interest are developer libraries (followed by `-dev`). Those contain among others

- header files (extension `.h`). Those constitute the interface for developers and are most often put in an include folder (e.g. `/usr/include`).
- library files (extension `.so`). Those provide the actual content of the library and are often put in a lib folder (e.g. `/usr/lib`).

To obtain the locations of such files, one can use the unix command (from the `dpkg` package)

```
dpkg -L [package]
```

Prerequisites

The prerequisites are first discussed qualitatively, because the exact installation of these might depend on the exact linux distribution. A possible list (tested for Ubuntu 19.10) is found at the end of this section.

1. Ubuntu or a similar linux distribution
2. A fortran compiler. Possible options:
 - gfortran, the GNU compiler. It is open source and available via e.g. apt (gfortran package).
 - ifort, the intel compiler. This is a commercial compiler, slightly more complicated to use but more optimized for intel architecture.
3. A MPI library installed (If you want to benefit from parallelism; recommended). Possible options:
 - Open MPI from apt (`libopenmpi-dev` package) or <http://www.open-mpi.org>
 - MPICH from apt (`libmpich-dev` package) or <http://www.mpich.org>

Depending on your linux distribution, you might need to manually add the

`mpi-default-dev` package, a metapackage for both MPI libraries.

1. A Linear Algebra library installed. A fallback (see next point) is available inside ABINIT (basic version of lapack), but you might want to install a math library yourself, especially for parallel computations: `blas` (`libblas-dev`), `lapack` (`liblapack-dev`), `scalapack` (`libscalapack-...-dev`), `atlas` (`libatlas-base-dev`), `mk1` (from Intel, or you might try the `libmkl-full-dev` package) ...
2. Some mandatory libraries:

- HDF5, NetCDF and NetCDF-Fortran, libraries to process binary files. Those are e.g. available via the `libhdf5-dev`, `libnetcdf-dev` and `libnetcdf-f-dev` packages from apt. For parallel IO also the `libpnetcdf-dev` is required.
- LIBXC, a library containing exchange-correlation potentials, e.g. from the `libxc-dev` package.

It is also possible to generate these libraries via the ABINIT fallbacks:

```
cd fallbacks ./build-abinit-fallbacks.sh
```

In the latter case the ABINIT config file (see later) should contain

```
with_fallbacks="yes" .
```

Following code installs a possible selection of required packages from apt for a simple parallel ABINIT compilation. This list might depend on your linux distribution, the exact ABINIT version you want to compile and the libraries you want to use.

```
# 1 # compiler
sudo apt install gfortran

# 2 # MPI libraries - choice for Open MPI
sudo apt install mpi-default-dev libopenmpi-dev

# 3 # math libraries - choice for lapack and blas
sudo apt install liblapack-dev libblas-dev

# 4 # mandatory libraries
sudo apt install libhdf5-dev libnetcdf-dev libnetcdf-f-dev libpnetcdf-dev libxc-
```

Installing ABINIT

Download ABINIT. For normal users it is advised to get the newest version from our website

(replace 9.0.4 by the newest version available).

```
wget https://www.abinit.org/sites/default/files/packages/abinit-9.0.4.tar.gz
tar xzf abinit-9.0.4.tar.gz
cd abinit-9.0.4
```

Create a working directory:

```
mkdir build && cd build
```

To configure:

```
../configure --with-config-file='my_config_file.ac9'
```

where 'myconfigfile.ac9' is either a self made config file. More on the configure options is presented in next section.

Compile with:

```
make mj4
```

Install (optional):

```
make install
```

The config file

The configure command takes as input some variables and flags. For example

```
../configure FC=mpifort --with-mpi="yes"
```

tells ABINIT to configure for a compilation with gfortran and to enable MPI support. Any variable or flag can be found by typing

```
../configure --help
```

Most options are detected automatically by ABINIT. For example, if `with_mpi` is set to 'yes', ABINIT will try to use the parallel fortran compiler (mpifort) and detect directories with useful library (.so) and header (.h) files for MPI support. When you installed the Open MPI

package via apt, these directories can be displayed by using

```
dpkg -L 'libopenmpi-dev' .
```

When a lot of options are used, it is advised to use a config file. For example, a parallelized version of abinit using lapack and blas is obtained by using the config file

```
# MPI settings
with_mpi="yes"
enable_mpi_io="yes"

# linear algebra settings
with_linalg_flavor="netlib"
LINALG_LIBS="-L/usr/lib/x86_64-linux-gnu -llapack -lblas"

# mandatory libraries
with_libxc="yes"
with_hdf5="yes"
with_netcdf="yes"
with_netcdf_fortran="yes"
```

Remark that

- one uses '-' when typing a flag but '_' inside the config file, e.g. `--with-mpi="yes"` becomes `with_mpi="yes"`.
- the LINALG_LIBS variable was explicitly set for this linux distribution. The directory was extracted via `dpkg -L liblapack-dev` and `dpkg -L libblas-dev`.
- when finetuning variables and flags for a particular linux distribution, it is advised to take a look at the template file `~abinit/doc/build/config-template.ac9`. For example, the setting of LINALG_LIBS in this template file is given by the line `#LINALG_LIBS="-L/usr/local/lib -llapack -lblas"`.

More specialized libraries might be harder to detect. For example, following section was added to the config file to detect a customized FFT and XML library. These libraries are available via apt (`libfftw3-dev` and `libxml2-dev`). The directories for the corresponding library and header files can be found by using `dpkg -L [package]` and other flags can be extracted from the `~abinit/doc/build/config-template.ac9` template

```
# fast fourier settings
with_fft_flavor="fftw3"
FFTW3_CPPFLAGS="-I/usr/include"
FFTW3_FCFLAGS="-I/usr/include"
FFTW3_LIBS="-L/usr/lib/x86_64-linux-gnu -lfftw3 -lfftw3f"

# XML settings (necessary for multibinit)
with_libxml2="yes"
LIBXML2_FCFLAGS="-I/usr/lib/x86_64-linux-gnu/"
LIBXML2_LIBS="-L/usr/include/libxml2/libxml/ -lxmlf90"
```