

EasyBuild hackathon @ Cyprus (March 11-13 2013)

EasyBuild training exercises

Kenneth Hoste and Jens Timmerman
kenneth.hoste@ugent.be – jens.timmerman@ugent.be

These exercises are intended to give you some basic hands-on experience with EasyBuild, in preparation of an EasyBuild hackathon.

If you have problems or questions, do not hesitate to contact us (email, IRC, ...)!

0) Warmup

Go to the main EasyBuild website <http://hpcugent.github.com/easybuild/> and from there:

- Subscribe to the EasyBuild mailing list, so you can stay informed about EasyBuild (feel free to unsubscribe again later).
- Create a personal GitHub account (free) and start watching and/or 'star' the three main EasyBuild repositories.
- Find the list of currently supported software packages on the EasyBuild wiki. Are there any packages there that are of interest to you?

1) Installing EasyBuild

Before we can actually get some hands-on experience with EasyBuild, we need to get it installed. Make sure the required EasyBuild dependencies (Linux, Python 2.4 or more recent 2.x, and environment modules) are fulfilled.

The easiest way to get EasyBuild installed is using the bootstrap procedure documented at

<https://github.com/hpcugent/easybuild/wiki/Bootstrapping-EasyBuild>

The bootstrap procedure will yield an EasyBuild module that you can load, just like for any other software package. It doesn't require admin rights on the system since it installs EasyBuild with EasyBuild (but you shouldn't worry about that).

To check whether the bootstrap procedure worked, load the EasyBuild module and check the version:

Listing 1: Commands to verify EasyBuild bootstrap procedure.

```
module load EasyBuild
eb --version
```

Note: The bootstrap procedure is quite new, and although it has been well tested on different systems, it may fail to work for you. In case that happens, please contact the EasyBuild team with a description of your problem, so it can be resolved for others. As a fallback, you can install EasyBuild manually following the guidelines at <https://github.com/hpcugent/easybuild/wiki/Installing-EasyBuild>.

2) EasyBuild command line

Working with EasyBuild is a synonym to using the `eb` command. The following questions/tasks below should make you more familiar with how it works (pro tip: `eb --help`).

- a) How do you enable debug log messages?
- b) What are the supported values for the `--stop` command line option?
- c) How many compiler toolchains does EasyBuild v1.2.0 support out-of-the-box? Which compiler does each toolchain employ?
- d) Which values are compulsory in `easyconfig (.eb)` files?
- e) Try and generate a `.dot` file with the dependency graph for WRF v3.3.1 (ictce toolchain).

3) Building and installing software

Building and installing software with EasyBuild is easy (heh!). To get a feel, try completing the following basic tasks:

- a) Install ABINIT v7.0.3 (binary release). (tip: the `--search` command line option can be useful here)
- b) Install ABINIT v7.0.4 (binary release). (tip: see the `--try-X` software build options in `--help`)
- c) Build and install `gzip` using a dummy toolchain (i.e., using the system compilers).
If you need help, see <https://github.com/hpcugent/easybuild/wiki/Getting-started>.
- d) Build and install the `gompi` (GCC+OpenMPI) toolchain.
Note: this may take a while to build, about 1h on a recent system.
- e) Build and install `gzip` using the `gompi` toolchain.
- f) Build and install the `goalf` (GCC+OpenMPI+ATLAS+LAPACK+FFTW) toolchain.
Note: due to ATLAS this may take very long (15m up to 10h and more, depending on your system).

4) Software packages of interest

Compose a list of software packages for which you'd like to have support for in EasyBuild.

For inspiration, see <https://github.com/hpcugent/easybuild-easyblocks/issues>, e.g., :

- PRACE packages: <https://github.com/hpcugent/easybuild-easyblocks/issues/82>
- PETTT ACE packages: <https://github.com/hpcugent/easybuild-easyblocks/issues/85>
- HPC-BC packages: <https://github.com/hpcugent/easybuild-easyblocks/issues/86>
- Perl: <https://github.com/hpcugent/easybuild-easyblocks/issues/92>

5) Building a not yet supported package (advanced)

Although EasyBuild already supports a significant amount of software packages, there is a lot more out there. Up to a certain point, it is possible to automatically generate easyconfig files from other packaging tools, e.g. pkgsrc.

Task: find a software package listed in <https://github.com/fgeorgatos/easybuild.experimental/tree/master/contrib/pkgsrc/20121226> that is not yet available in EasyBuild (see <https://github.com/hpcugent/easybuild-easyconfigs/tree/develop/easybuild/easyconfigs>).

Clean up the easyconfig file (.eb) for that software package (omit informative comments, put a decent list of sanity check paths in place, etc.), and try to make the software package build and install with EasyBuild.

A list of software packages that (seemed) to build correctly from an automatically generated easyconfig file is available at https://github.com/fgeorgatos/easybuild.experimental/blob/master/contrib/pkgsrc/20121226/README_delivered_modules.txt.

6) More advanced: easyblocks and toolchains

For software with more complex build procedures, an easyblock needs to be implemented. See <https://github.com/hpcugent/easybuild/wiki/Tutorial%3A-building-WRF-after-adding-support-for-it> for a tutorial example for WRF.

What adding support for a new compiler toolchains involves, is described at <https://github.com/hpcugent/easybuild/wiki/Compiler-toolchains>.