

EasyBuild reference card

<http://hpcugent.github.io/easybuild/>
<https://github.com/hpcugent/easybuild/wiki>

Getting started

- 1) Visit website: <http://hpcugent.github.io/easybuild/>
- 2) Subscribe to mailing list, to obtain emerging info
- 3) Create GitHub account & watch/star 4 EB repos
- 4) Find EasyBuild wiki → list of supported apps
- 5) Ensure: `GCC>=any AND Python>=2.4`
- 6) Ensure: `env-modules>=3.2.10 OR lmod>=5.1.5`
- 7) Install EasyBuild using bootstrapping procedure:
`python bootstrap_eb.py $HOME/.local/easybuild`
see <easybuild/wiki/Bootstrapping-EasyBuild>
- 8) Run unit tests: `python -m test.framework.suite`
- 9) `eb --version` # check you are up to speed
- 10) `eb --help` # list available options - *READ THIS*

Informational Options

```
eb --search=STR # search easyconfigs repo for STR
eb --search=/R- # search easyconfigs for R only
eb --list-easyblocks # lists easyblock types
eb --avail-easyconfig-constants # as described
eb -a #as described- SPEND TIME READING THIS
eb --show-default-moduleclasses # categories to
be used for moduleclasses (does not affect builds)
eb --list-toolchains # as described - CHECK IT
eb --dep-graph=degraph.<ext> # make .dot, .png
or other file with the graph of a particular build tree
```

Example run

```
$ eb FCM-2.3.1.eb
== temporary log file in case of crash
/tmp/3L/3LkBznPF7WgIo752F4GjE+++TI-
Tmp-/easybuild-A2VKzN.log
== resolving dependencies ...
== processing EasyBuild easyconfig
/Users/fotis/Desktop/arena/uni.lu/easybuild-
easyconfigs/easybuild/easyconfigs/f/FCM/FCM-
2.3.1.eb
== building and installing FCM-2.3.1...
== fetching files...
== creating build dir, resetting environment...
== unpacking...
== patching...
== preparing...
== configuring...
== building...
== testing...
== installing...
== taking care of extensions...
== packaging...
== postprocessing...
== sanity checking...
== cleaning up...
== creating module...
== COMPLETED: Installation ended successfully
== Results of the build can be found in the log file
/tmp/3L/3LkBznPF7WgIo752F4GjE+++TI-
Tmp-/easybuild-FCM-2.3.1-20130915.104258.log
== Build succeeded for 1 out of 1
== temporary log file
/tmp/3L/3LkBznPF7WgIo752F4GjE+++TI-
Tmp-/easybuild-A2VKzN.log has been removed.
```

Build a compiler toolchain (pick one)

```
eb --try-software-name=goolf -r # build (~1hr)
toolchain of OSS components, based on OpenBLAS
eb --try-software-name=goalf -r # build (>1hr)
toolchain of OSS components, based on ATLAS; this
may need debug/tuning in VMs, special nodes etc.
eb --try-software-name=ictce -r # install
toolchain consisting of Intel Compilers, Intel MPI
stack, Intel MKL, etc. - requires sources & license!
eb --try-software-name=cgmvolf -r # build
toolchain with Clang for C/C++, GCC for Fortran,
MVAPICH, OpenBLAS, (Sca)LAPACK, FFTW; i.e.
this is a drop-in replacement for goolf, goalf or ictce
eb --try-software-name=goolfc -r # build stack
similar to goolf, yet include CUDA in the toolchain
eb --try-software-name=gompi -r # build
toolchain with only GCC/OpenMPI; part of go(o)a)lf
```

Sample builds

```
eb --try-software-name=ABINIT # install pre-
built ABINIT, no compiling done (TarBall easyblock)
eb CMake-2.8.4-goolf-1.4.10.eb -r # install a
version of CMake (ConfigureMake easyblock)
eb gzip-1.5-goolf-1.4.10.eb --try-software-
version=1.6 --try-toolchain-name=ictce -r
# attempt to build a more recent gzip version using
the goolf easyconfig as template, using most recent
ictce compiler toolchain (i.e., Intel tools)
eb VTK-5.10.1-goolf-1.4.10.eb -r # install VTK
with its regular procedure (CMakeMake easyblock)
eb biodeps-1.6-goolf-1.4.10.eb -r # install
biodeps module, providing common dependencies
eb --try-software-name=wiki2beamer --try-
toolchain=goolf,1.4.10 -r # build wiki2beamer,
using goolf toolchain (PythonPackage easyblock)
eb BioPerl-1.6.1-goolf-1.4.10-Perl-5.16.3.eb
-r # build BioPerl v1.6.1 (PerlModule easyblock)
```

Picking up experience

```
eb R-2.15.2-goolf-1.4.10.eb -r # install a
version of R - requires Java, must be available
eb GROMACS-4.6.1-goolfc-1.3.12.eb -r # install
GROMACS against CUDA-aware goolf toolchain (!)
eb WRF-3.3.1-goolf-1.4.10-dmpar.eb # build
WRF along all its dependencies - this is a long one!
incl. netCDF(-Fortran), HDF5, custom Doxygen...
eb petsc4py-3.3-goolf-1.4.10-Python-2.7.3.eb
-r # this includes PETSc, that brings-in many
extras: Python, Boost, FIAT, (Par)METIS, SciPy,
SCOTCH, Hypre, SuiteSparse (incl. CHOLMOD,
UMFPACK)
eb DOLFIN-1.0.0-goolf-1.4.10-Python-2.7.3.eb
--dry-run -r # overview of what will be installed;
building it may be tricky, due to many dependencies
```

Kudos to UGent HPC team for providing EasyBuild as Open Source.
Kudos to FOSSwire for the original template for this cheatsheet:
<http://fosswire.com/post/2007/08/unixlinux-command-cheat-sheet/>
Page compiled in CC-BY-SA terms by Fotis Georgatos <fotis@cern.ch>,
with feedback from Kenneth Hoste <kenneth.hoste@ugent.be>
Kindly address feedback to each as needed:

- software feedback should go to GitHub repos
- cheatsheet feedback go to declared author



HPCBIOS policies scope

HPCBIOS is an effort to setup a common, defined, well-documented and, reproducible environment spanning across multiple HPC systems and sites, with a special focus on Life Science applications.

HPC Baseline Configuration includes:

HPCBIOS_05-01: Multiple-Version Software Policy
HPCBIOS_05-05: Common Queue Names
HPCBIOS_05-06: Baseline Set of Login Shells
HPCBIOS_06-01: Common Set of Open Source Math Libraries
HPCBIOS_06-04: Baseline Editors and Scripting Tools
HPCBIOS_06-05: Baseline Set of Debuggers
HPCBIOS_06-15: Sample Code Repository
HPCBIOS_06-17: Use Modules for Accessing Multiple Versions of Software
HPCBIOS_06-19: Common Set of Open Source Utilities
HPCBIOS_07-02: Common Open Source Performance and Profiling Tools
HPCBIOS_07-03: Common Set of Open Source Compilers
HPCBIOS_10-01: New/Returning User Welcome Letter
HPCBIOS_10-02: Common Open Source High Productivity Languages
HPCBIOS_2012-80: Common Set of DFT codes
HPCBIOS_2012-90: Software Tools and Development Environment
HPCBIOS_2012-91: Modules Namespace for HPC sites
HPCBIOS_2012-92: EasyBuild HPC Software Development Environment
HPCBIOS_2012-93: Life Sciences Productivity Environment
HPCBIOS_2012-94: Bioinformatics & Comp. Biology Productivity Env/ment
HPCBIOS_2012-95: Molecular Dynamics Productivity Environment
HPCBIOS_2012-96: Common Set of Commercial Compilers
HPCBIOS_2012-97: Climate Science Productivity Environment
HPCBIOS_2012-98: Common Set of Environment Variables
HPCBIOS_2012-99: GPU Productivity Environment
HPCBIOS_2013-01: Common Dependencies for Life Science Applications

HPCBIOS list of policies are work in progress

- there is no obligation for any HPC site to follow all stated targets; only to document status precisely.

Getting the basic services

```
module load EasyBuild/1.7.0 # or greater
eb --search HPCBIOS - detect what is available
eb HPCBIOS_Bioinfo-20130829-ictce-5.3.0.eb -r
# build Bioinformatics tools w. Intel
eb HPCBIOS_Bioinfo-20130829-goolf-1.4.10.eb
-r # build Bioinformatics tools w. GNU
eb HPCBIOS_LifeSciences-20130829-goolf-
1.4.10.eb -r # build LifeSciences w. Intel
eb HPCBIOS_LifeSciences-20130829-ictce-
5.3.0.eb -r # build LifeSciences w. GNU
eb HPCBIOS_Debuggers-20130829.eb -r # deliver
debuggers: GDB, TotalView/MemoryScape, IDB...
eb HPCBIOS_Math-20130829-goalf-1.1.0.eb -r #
deliver GCC, OpenMPI, ATLAS, FFTW, PETSc, GSL
eb HPCBIOS_Math-20130829-goolf-1.4.10.eb -r #
same as goalf but replace ATLAS with OpenBLAS
eb HPCBIOS_Math-20130829-ictce-5.3.0.eb -r #
deliver icc/ifort, impi, imkl, PETSc, GSL
eb HPCBIOS_Profilers-20130829.eb # deliver
tools useful in profiling:
• VTune/2013_update10
• Inspector/2013_update6
• itac/8.0.0.011
• PAPI/5.0.1
• Valgrind/3.8.1
• binutils/2.22 # provides gprof/2.22
```

```
eb PRACE-20130605-goolf-1.4.10.eb -r # build
Bash/tcsh/make, Tcl/Tk, netCDF, Perl, Java...
eb PRACE-20130605-ictce-5.3.0.eb -r # build
Bash/tcsh/make, Tcl/Tk, netCDF, Perl, Java...
eb PRACE-ENV-20130605.eb # provides $PRACE_*
FLAGS, CFLAGS, LDFLAGS, STORE, SCRATCH...
eb biodeps-1.6-*.*.eb - ie. build HPCBIOS_2013-01
```

Custom installation of packages

```
$ time eb --try-toolchain=ictce,5.3.0 \
--try-software-name=jellyfish --try-software-version=1.1.10 \
--try-amend=source_urls=http://www.ccbc.umd.edu/software/jellyfish \
--try-amend=sources=jellyfish-1.1.10.tar.gz # without any easyconfig file!!
== temporary log file in case of crash /tmp/easybuild-h2cCbR.log
== Generated an easyconfig file jellyfish-1.1.10-ictce-5.3.0.eb, going to
use it now...
== resolving dependencies ...
== processing EasyBuild easyconfig
/home/users/homedirs/fgeorgatos/jellyfish-1.1.10-ictce-5.3.0.eb
== building and installing jellyfish-1.1.10-ictce-5.3.0...
== fetching files...
== creating build dir, resetting environment...
== unpacking...
== patching...
== preparing...
== configuring...
== building...
== testing...
== installing...
== taking care of extensions...
== packaging...
== postprocessing...
== sanity checking...
== cleaning up...
== creating module...
== COMPLETED: Installation ended successfully
== Results of the build can be found in the log file /tmp/easybuild-
jellyfish-1.1.10-20130915.212339.log
== Build succeeded for 1 out of 1
== temporary log file /tmp/easybuild-h2cCbR.log has been removed.
```

```
real    0m17.309s
user    0m29.562s
sys     0m7.652s
```

```
$ eb --try-software-name=Maq
--try-software-version=0.7.1
--try-toolchain=goalf,1.1.0-no-OFED
--try-amend=sources=maq-0.7.1.tar.bz2 --try-
amend=source_urls=http://sourceforge.net/pro
jects/maq/files/maq/0.7.1 - install MAQ using
goalf toolchain; N.B. You will need to clean this up:
```

- Remove redundant comments
- Fix headers to include correct pointers
- Set up sanity checks correctly
- Verify that version is defined as an %s
construct in the source blob,
to allow --try-software-version feature
- Perhaps try more toolchains, as applicable

Success? - It's time to contribute back via [GitHub!](#)

Hints

Bootstrapping EasyBuild - simply follow:

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

Repository for draft easyconfigs - testing OK:

<https://github.com/fgeorgatos/easybuild.experimental/>

Experimental easyconfigs from pkgsrc - drafts:

<https://github.com/fgeorgatos/easybuild.experimental/tree/master/contrib/pkgsrc/20130506> - README_delivered_modules.txt # successful ones

Search and report issues - in the right place:

- <https://github.com/fgeorgatos/HPCBIOS/issues>
- <https://github.com/fgeorgatos/easybuild.experimental/issues>
- <https://github.com/hpcugent/easybuild-framework/issues>
- <https://github.com/hpcugent/easybuild-easyblocks/issues>
- <https://github.com/hpcugent/easybuild-easyconfigs/issues>

eb -ld ... - show full debug log during build

eb --stop <where> - stop at step <where>

Your own easyblocks repository - HOWTO:

<https://github.com/hpcugent/easybuild/wiki/Setting-up-your-own-
easyblocks-repository>

easyblocks & easyconfigs tutorial:

<https://github.com/hpcugent/easybuild/wiki/Tutorial%3A-building-WRF-
after-adding-support-for-it>