

*building software with ease*

ISC'14 BoF lightning talk  
Getting Scientific Software Installed: Tools & Best Practices  
June 24th 2014

*[jens.timmerman@ugent.be](mailto:jens.timmerman@ugent.be)  
[easybuild@lists.ugent.be](mailto:easybuild@lists.ugent.be)*

# “Please install this software on the cluster?”

Scientists focus on the *science* of the software they produce, not on build procedure, portability, ...

This makes building/installing (lots of) scientific software **painful**:  
*very time-consuming, error-prone, hard to get right, ...*



Common issues:

- ▶ non-standard build tools
- ▶ incomplete build procedure  
e.g., no install step
- ▶ interactive scripts
- ▶ hardcoded parameters
- ▶ poor/outdated documentation
- ▶ ...

***Lots of duplication of work across HPC sites!***

# EasyBuild: building software with ease



<http://hpcugent.github.io/easybuild>

EasyBuild is a *software build and installation framework*.

- ▶ written in **Python**
- ▶ started in 2009, in-house for ~2.5 years, **GPLv2** since 2012
- ▶ **stable API** since EasyBuild v1.0 (Nov'12), latest is v1.13.0
- ▶ continuously enhanced and extended, thoroughly tested
- ▶ *release early, release often strategy* (major version every 4-6 weeks)
- ▶ development is highly **community-driven**



# EasyBuild community

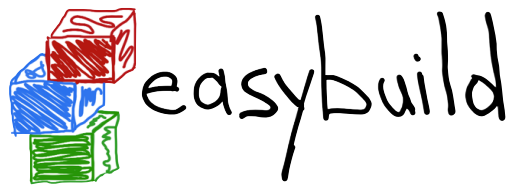


*6th EasyBuild hackathon*

*June 18-20th 2014, Vienna (Austria)*

Ghent University & VSC sites (Belgium)  
University of Luxembourg  
Gregor Mendel Institute (Austria)  
The Cyprus Institute  
University of Basel (Switzerland)  
Jülich Supercomputer Centre (Germany)  
Bayer (Germany)  
University of Auckland (New Zealand)  
Kiev Polytechnic Institute (Ukraine)  
Idaho National Lab (US)  
Pacific Northwest National Lab (US)  
UC Davis (US)  
& (many?) more...

+ support from NVIDIA, TACC (Lmod), ...



# Using EasyBuild

0) Easily install EasyBuild by bootstrapping it:

```
$ curl -O http://hpcugent.github.io/easybuild/bootstrap_eb.py  
$ python bootstrap_eb.py <prefix>
```

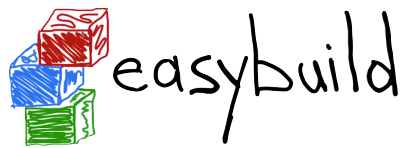
1) Set module path, load EasyBuild module, basic configuration:

```
$ export MODULEPATH=<prefix>/modules/all:$MODULEPATH  
$ module load EasyBuild  
$ export EASYBUILD_PREFIX=<prefix>
```

**2) Example: build WRF & all deps using Intel compilers/libraries:**

```
$ eb WRF-3.5-ictce-5.3.0.eb --robot  
$ module av WRF  
WRF/3.5-ictce-5.3.0
```

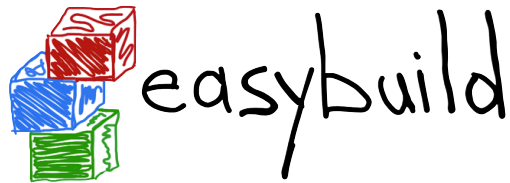




# List of supported software (v1.13.0)

*483 different software packages (2,501 example builds)*

a2ps ABAQUS ABINIT ABySS ACML **ALADIN** Alinea ALLPATHS-LG AMOS AnalyzeFMRI ant APBS ARB argtable aria2 Armadillo arpack-ng ASE ATLAS Autoconf Automake bam2fastq BamTools Bash BayesTraits bbcp bbFTP bbftpPRO bc beagle-lib BEDTools BFAST binutils BioPerl Biopython BiSearch Bison BitSeq BLACS BLAST BLAT BOINC Bonnie++ Boost Bowtie Bowtie2 BWA byacc bzip2 cairo CAP3 CBLAS ccache CCfits CD-HIT CDO CEM CFITSIO cflow CGAL cgdb Chapel CHARMM Clang CLHEP CLooG Clustal-Omega ClustalW2 CMake Coreutils Corkscrew **CP2K** CPLEX CRF++ Cube CUDA Cufflinks cURL cutadapt CVS CVXOPT Cython DB Diffutils DL\_POLY\_Classic Docutils **DOLFIN** Doxygen **EasyBuild** ECore ed Eigen ELinks ELPA EMBOSS EPD ErlangOTP ESMF ESPResSo expat eXpress FASTA fasthack FASTX-Toolkit FCM FDTD\_Solutions Ferret FFC FFTW FIAT file findutils fixesproto flex FLTK FLUENT fmri FoldX fontconfig FRC\_align freeglut FreeSurfer freetype FSL g2clib g2lib GATE GATK gawk GCC gccuda GDAL GDB Geant4 GEMSTAT GenomeAnalysisTK GEOS gettext GHC Ghostscript GIMPS git GLib GLIMMER GLPK glproto GMAP GMP GMT gnuplot gnutls Go google-sparsehash GPAW gperf gperftools Greenlet grep grib\_api GROMACS GSL GTI guile gzip h4toh5 h5py h5utils Harminv HDF HDF5 HH-suite HMMER horton HPL HTSeq hwloc Hypre icc ifort imake imkl impi Infernal inputproto Inspector Instant lperf ipp IPython Isolnfer ispc itac Jansson JasPer Java Jellyfish Jinja2 JUnit kbproto LAPACK less lftp libcircle libctl libdrm libffi libgtextutils libharu libibmad libibumad libibverbs libICE libidn Libint libint2 libjpeg-turbo libmatheval libpciaccess libpng libpthread-stubs libreadline libSM libsmm LIBSVM LibTIFF libtool libungif libunistring libunwind libX11 libXau libXaw libxc libxcb libXext libXfixes libXi libxml2 libXmu libXp libXpm libxslt libXt libyaml likwid Lmod Lua LWM2 lxml lynx LZO M4 MAFFT make makedepend Maple MariaDB Mathematica MATLAB matplotlib mc MCL mcpp MDP Meep MEME Mercurial Mesa Mesquite MetaVelvet METIS MMSEQ Molden Molekel molmod Mothur motif MPFR mpi4py mpiBLAST MPICH MPICH2 MrBayes MTL4 MUMmer MUMPS MUSCLE MUST MUSTANG MVAPICH2 nano NASM NCBI-Toolkit ncd4 **NCL** ncurses NEdit netaddr netCDF netCDF-C++ netCDF-Fortran netcdf4-python netifaces netloc nettle **NEURON** nodejs ns numactl numexpr numpy NWChem O2scl Oases OCaml Oger OPARI2 OpenBabel OpenBLAS **OpenFOAM** **OpenFOAM-Extend** OpenIFS OpenMPI OpenPGM OpenSSL ORCA orthomcl otcl OTF OTF2 packmol PAML pandas PANDAsseq PAPI parallel Paraview ParFlow ParMETIS ParMGridGen Pasha patch paycheck PCC PCRE PDT Perl PETSc petsc4py phonopy PhyML picard pixman pkg-config PLINK PnMPI popt PP PRACE Primer3 printproto problog protobuf pscom PSI psmpi2 PyQuante pysqlite pyTables **Python** python-dateutil python-meep PyYAML PyZMQ QLogicMPI Qt qtop QuadProg++ **QuantumESPRESSO** R RAXML RCS RNAz ROOT Rosetta rSeq RSEQtools Ruby Sablotron SAMtools ScaLAPACK Scalasca ScientificPython scikit-learn scipy SCons SCOOP Score-P SCOTCH SDCC sed segemehl setuptools Shapely SHRIMP sickle Silo slalib-c SLEPc SOAPdenovo SOAPdenovo2 SOAPec Sphinx SQLite SRA-Toolkit Stacks Stow Stride SuiteSparse SURF SWIG sympy Szip TAMkin Tar tbb TCC Tcl tccl tcsh Tesla-Deployment-Kit texinfo Theano TiCCutils TiMBL TinySVM Tk TopHat Tornado TotalView TREE-PUZZLE **Trilinos** Trinity UDUNITS UFC UFL util-linux Valgrind VCFtools Velvet ViennaRNA Vim Viper vsc-base vsc-mypirun vsc-mypirun-scoop vsc-processcontrol VSC-tools VTK VTune **WIEN2k** wiki2beamer **WPS** **WRF** xbitmaps xcb-protocol XCrySDen xextproto XML XML-LibXML XML-Simple xorg-macros xproto xtrans XZ yaff YamCha YAML-Syck Yasm ZeroMQ zlib zsh zsync



# Current features, future directions

currently (EasyBuild v1.13.0):

- supports GCC/Clang/Intel compilers & various MPI/BLAS/LAPACK/FFT libraries
- robust **framework** providing supporting functionality
- very **dynamic design**: plugin support for new compiler/MPI/software package
- **generates module files** (Tcl), supports using (C/)Tcl & Lmod module tools
- support for using a *custom module naming scheme* you define yourself
- **fully autonomous builds**, build logging, automagical dependency resolution, ...

work in progress:

- *support for hierarchical module naming schemes*
- improve integration with Lmod (e.g. support for generating Lua module files)
- better support for site customization
- better error reporting
- support for more compilers, software applications and platforms
- new format for easyconfig files (build specs) to handle explosion of contributions

**EasyBuild community drives most new features, so *get involved!***



# easybuild

*building software with ease*



Do you want to know more?

**website:** <https://hpcugent.github.io/easybuild>

**GitHub:** [https://github.com/hpcugent/easybuild\[-framework\]-easyblocks\[-easyconfigs\]](https://github.com/hpcugent/easybuild[-framework]-easyblocks[-easyconfigs])

**PyPi:** [https://pypi.python.org/pypi/easybuild\[-framework\]-easyblocks\[-easyconfigs\]](https://pypi.python.org/pypi/easybuild[-framework]-easyblocks[-easyconfigs])

**mailing list:** [easybuild@lists.ugent.be](mailto:easybuild@lists.ugent.be)

**Twitter:** [@easy\\_build](https://twitter.com/easy_build)

**YouTube:** search for “EasyBuild intro”,  
“EasyBuild WRF”

**IRC:** #easybuild on freenode.net

