



LCG software stack management

Benedikt Hegner (CERN)

HEP Software Foundation Workshop
20.1.2015

LCG software stack

- The LCG software stack is a collection of > 100 software packages from HEP and other communities tested on ~ 10 OS * compiler combinations
- Provided to ATLAS, LHCb and various smaller experiments
- Based on the following four independent choices:
 - CMake eco-system (CMake, CTest, CDash) for building
 - Jenkins for nightly and release build automatization
 - RPM and CVMFS for distribution
 - auto-generated shell scripts for setting consistent environment
- The LCG Generator Services project has been merged into this system
- Could be generalized to be useful for other projects/experiments as well

<http://lcgsoft.web.cern.ch/lcgsoft/>

CMake as build infrastructure

- get or setup **cmake**
- checkout **lcgcmake** package from SVN
- setup C/C++/Fortran compilers
- create workspace area
- configure with **cmake**
- build with **make**

1. On lxplus set PATH to use one of latest CMake versions (default is 2.6)
`export PATH=/afs/cern.ch/sw/lcg/contrib/CMake/2.8.12.2/Linux-i386/bin:${PATH}`
2. Checkout the lcgcmake package from lcgsoft SVN repository
`svn co svn+ssh://svn.cern.ch/repos/lcgsoft/trunk/lcgcmake`
3. Create a workspace area in which to perform the builds
`mkdir lcgcmake-build`
`cd lcgcmake-build`
4. You may need at this moment to define the compiler to use if different from the native compiler
`source /afs/cern.ch/sw/lcg/external/gcc/version/platform/setup.(c)sh`
5. Configure the build of all externals with cmake
`cmake -DCMAKE_INSTALL_PREFIX=../lcgcmake-install ../lcgcmake`
6. In order to build against the existing external repository use the option
`-DLCG_INSTALL_PREFIX=/afs/cern.ch/sw/lcg/external`
to tell the system to look for packages in the LCG area. Other available options are: `-DLCG_VERSION=XX` to select a given LCG configuration version, `-DLCG_IGNORE='package1;package2;...'` to ignore packages that are already existing in LCG area and force a re-build.
7. Build and install all external packages
`make -jN`
8. Or to build a single external package (make help shows all possibilities)
`make -jN <package>`
9. You may need to restart the build of a package from beginning in case of obscure errors. The best is to clean a specific package
`make clean-<package>`

Defining Packages and Stacks

```
LCGPackage_Add( pythia8
  URL ~torbjorn/pythia8/pythia8<NATIVE_VERSION>.tgz
  CONFIGURE_COMMAND ./configure --prefix=<INSTALL_DIR>
    --with-hepmc=${HepMC_home}
    --enable-shared
  BUILD_IN_SOURCE 1
  CONFIGURE_EXAMPLES_COMMAND <SOURCE_DIR>/examples/configure
    --with-pythia8=<INSTALL_DIR>
    --with-lhapdf=${lhpdf_home}
  BUILD_EXAMPLES_COMMAND make -C <SOURCE_DIR>/examples main01 main03
  INSTALL_EXAMPLES_COMMAND cmake -E make_directory <INSTALL_DIR>/bin
  COMMAND cmake -E chdir <SOURCE_DIR>/examples
    cp bin/main01.exe bin/main03.exe main03.cmd <INSTALL_DIR>/bin
  TEST_COMMAND cmake -E chdir <INSTALL_DIR>/bin ./main01.exe
  DEPENDS HepMC lhpdf
)
```

```
...
LCG_external_package(graphviz          2.28.0      )
LCG_external_package(GSL                1.10        )
LCG_external_package(HepPDT             2.06.01    )
LCG_external_package(ipython            0.12.1     )
LCG_external_package(json               2.5.2      )
LCG_external_package(lapack              3.5.0      )
LCG_external_package(lcov                1.9         )
LCG_external_package(libsvm              2.86       )
LCG_external_package(libtool             1.5.26     )
LCG_external_package(lxml                2.3        )
LCG_external_package(matplotlib         1.3.1      )
LCG_external_package(minuit              5.27.02   )
...
```

Multiple stacks supported at the same time