



# Ideas on Software Distribution Model

Librarians and Integrators Meeting  
(4.6.2013)

Benedikt Hegner, Pere Mato

# New CMake build infrastructure

Fairly simple solution for building external now

- 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/external/CMake/2.8.9/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 lcgmak-build`  
`cd lcgmak-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=../lcgmak-install ../lcgmak`
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.
7. Build and install all external packages  
`make -j`
8. Or to build a single external package  
`make -j <package>` (use `make help` to see the list of all available packages)
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>`

# New CMake infrastructure

- New release build infrastructure tested every night
  - Results shown on [cdash.cern.ch](http://cdash.cern.ch)
- Outstanding problems:
  - CORAL and COOL aren't integrated yet
  - The nightly summary page isn't integrated here

LoginAll Dashboards

Sunday, May 05 2013 21:34:08 CEST

LCGSoft

Dashboard

Calendar

Previous

Current

Project

No file changed as of Sunday, May 05 2013 - 03:00 CEST

Show FiltersSimple ViewAuto-refreshHelp

Experimental

Site	Build Name	Update		Configure		Build			Test				Build Time	
		Files	Time	Error	Warn	Time	Error	Warn	Time	Not Run	Fail	Pass		Time
macphsft20	x86_64-mac108-gcc42-opt ⓘ	0	0s	0	0	30s	50	1	1h 28m					1 hour
lcgapp07.cern.ch	x86_64-slc6-gcc48-opt ⓘ	0	0s	0	0	12s	4	1	1h 20m 42s					
lxbuild175.cern.ch	x86_64-slc5-gcc43-opt	0	6s	0	0	12s	0	0	1h 9m 42s					

A bit more to build

A bit more than 1 hour  
to build all externals  
and generators!



# Building Full Releases

- We propose a **new way** of providing releases
- Preparing the full stack is a much cheaper operation now
- Until a real release has been made, new build products will go into an experimental area only
  - the gcc48 build already follows this approach
  - Would like to apply the same to *all platforms* in the -preview slot soon!
- Once the release is cut, it will be made available in
  - CVMFS (for sure)
  - AFS (is that still required?)
- But how will such a release look like?



toolset.cmake



“release”



“platform”



“install\_type”



LCGCMake  
(cmake, make)





