GENSER overview

Mikhail Kirsanov and Dmitri Konstantinov
On behalf of the
GENSER SFT Team

GENSER

- The Generator Services project.
  - The purpose is to provide the following components for the LHC community and experiments:
    - installed MC generator libraries on network file systems (e.g. CERN AFS and CVMFS)
    - binary tarballs for remote sites
    - source tarballs with the tested codes (sometimes patched), for some generators with a build system from GENSER

Genser team (~ 1 FTE):

- Witold Pokorski (CERN, SFT)
- Dmitri Konstantinov (IHEP, Protvino)
- Mikhail Kirsanov (IHEP, Protvino)
- Grigory Gladyshev (IHEP, Protvino)

+ great help from Benedikt Hegner and Pere Mato Vila

Current status

- GENSER, previously based on bmake is fully migrated to cmake (using ExternalProject module) and integrated into LCG external build system(LCGSOFT):
  - saving resources and time;
  - dependencies resolution provided by LCG releases
  - automation;
  - aiming to produce new complete LCG releases with a single click;

Steps needed

- for a new generator:
  - add necessary instructions to generators/CMakeList.txt
  - download the tarfile to the GENSER local sources repository.
  - add new version modifying steering tool chain file.
- For a new LCG release:
  - prepare the new toolchain file with all requested packages and versions
  - build LCG externals and MC generators

Release packages and versions

- Package names and versions (externals + generators) are kept in one steering 'toolchain' file. A fragment:

```
#---Additional External packages------(Generators)-----------------

set(MCGENPATH  MCGenerators_lcgcmt${heptools_version})

LCG_external_package(powheg-box          r2092         ${MCGENPATH}/powheg-box    )
LCG_external_package(lhapdf              5.9.1         ${MCGENPATH}/lhapdf        )
LCG_external_package(lhapdf6             6.0.5         ${MCGENPATH}/lhapdf6        )
LCG_external_package(pythia8             175.lhetau    ${MCGENPATH}/pythia8       author=175     )
LCG_external_package(sacrifice           0.9.9         ${MCGENPATH}/sacrifice   pythia8=183 )
LCG_external_package(thepeg              1.9.0         ${MCGENPATH}/thepeg        )
LCG_external_package(herwig++            2.7.0         ${MCGENPATH}/herwig++      thepeg=1.9.0)
LCG_external_package(tauola++            1.1.4         ${MCGENPATH}/tauola++      )
LCG_external_package(pythia6             428.2         ${MCGENPATH}/pythia6         author=6.4.28 hepevt=10000  )
LCG_external_package(agile               1.4.1         ${MCGENPATH}/agile         )
LCG_external_package(photos++            3.54          ${MCGENPATH}/photos++      )
LCG_external_package(photos              215.4         ${MCGENPATH}/photos        )
```

Configuration and installation instructions are in generators/CMakeLists.txt

```
LCGPackage_Add(
  evtgen
  URL http://cern.ch/service-spi/external/tarFiles/MCGeneratorsTarFiles/evtgen-<evtgen_<NATIVE_VERSION>_tag>.tar.gz
  UPDATE_COMMAND <VOID>
  CONFIGURE_COMMAND ./configure --prefix=<INSTALL_DIR>
                                --hepmcdir=${HepMC_home}
                                --pythiadir=<pythia8-<evtgen_<NATIVE_VERSION>_pythia8>_home>
                                --photosdir=${photos++_home}
                                --tauoladir=<tauola++-<evtgen_<NATIVE_VERSION>_tauola++>_home>
  BUILD_COMMAND ${MAKE} -j1 "${evtgen-build-options}"
  INSTALL_COMMAND make install
          COMMAND ${CMAKE_COMMAND} -E chdir <INSTALL_DIR>/../share ${CMAKE_COMMAND} -E create_symlink  sources/evt.pdl
evt.pdl
          COMMAND ${CMAKE_COMMAND} -E chdir <INSTALL_DIR>/../share ${CMAKE_COMMAND} -E create_symlink  sources/
DECAY_2010.DEC DECAY.DEC
  BUILD_IN_SOURCE 1
  DEPENDS HepMC pythia8 photos++ tauola++
)
```

- A few lines in CMakeLists.txt required to describe steps to build a new packages
- Explicit handling of dependencies

Adding tests – 'add_test' in CMakeList.txt

```
#---test of the test infrasgtructure-----------------------------------
LCG_add_test(test-test PRE_COMMAND /bin/echo pre-comand 1
                       COMMAND /bin/echo pre-comand 2
                       COMMAND /bin/echo pre-comand 3
                       COMMAND /bin/echo pre-comand 4
                       COMMAND /bin/echo pre-comand 5
                  TEST_COMMAND /bin/echo test-command 1
                       COMMAND /bin/echo test-command 2
                       COMMAND /bin/echo test-command 3
                  POST_COMMAND /bin/echo post-command 1
                       COMMAND date )
```

Simple SHERPA test:

```
# $PWD is needed because Sherpa uses this variable to know current directory
LCG_add_test(sherpa_orig_test1
  PRE_COMMAND ${CMAKE_COMMAND} -E remove_directory sherpa/tests_orig1
      COMMAND ${CMAKE_COMMAND} -E make_directory sherpa/tests_orig1
  TEST_COMMAND ${CMAKE_COMMAND} -E chdir sherpa/tests_orig1 ${sherpa_home}/bin/Sherpa -j20 -f ${CMAKE_SOURCE_DIR}/generators/sherpa/tests/LHC_Z.dat
  ENVIRONMENT
    ${library_path}=${sherpa_home}/lib/SHERPA-MC:${HepMC_home}/lib:${lhapdf_home}/lib:${fastjet_home}/lib:${GSL_home}/lib:$ENV{${library_path}}
    PWD=.
)
```

Testing

- Tests run after each nightly/daily build
- All generators are covered by simple tests:
  - compile with library and run

- Work on the physics tests (get some results and check reproducibility on several platforms) is ongoing
- Rivet is widely used in these tests.

Testing started on 2014-03-31 11:49:25

Site Name: macitois17.cern.ch
Build Name: x86_64-mac108-clang34-opt
Total time: 16m 9s 640ms
OS Name: Mac OS X
OS Platform: x86_64
OS Release: 10.8.5
OS Version: 12F45
Compiler Version: unknown

7 tests failed.

Name	Status	Time	Details
evtgen-1.2.0_test1	Failed	1s 110ms	Completed (Failed)
rivet2_test_atlas	Failed	3s 280ms	Completed (Failed)
rivet2_test_atlas_script	Failed	3s 470ms	Completed (Failed)
rivet2_test_cms	Failed	6s 710ms	Completed (Failed)
rivet2_test_cms_script	Failed	2s 870ms	Completed (Failed)
sacrifice_test	Failed	300ms	Completed (Failed)
sherpa_orig_test1	Failed	1s 920ms	Completed (Failed)

Build instructions

- Requirements:
  - CMake 2.9
  - Set of compilers
- Simple steps:
  - checkout LCGSOFT from SVN
  - create build area
  - configure with cmake
  - build with ‘make’

How to build LCGSoft with CMake

The following are very basic instructions on how to build a few external packages to demostrate the new way of doing it using the ExternalProject module of CMake.

Please have a look at the svn repository for the lcgcmake package to see the different files involved in the procedure.

Practical instructions:

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/reps/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 -jN <package>` (use `make help`to see the list of all available packages)
9. You may need to restart de build of a package from beginning in case of obscure errors. The best is to clean a specific package
   `make clean-<package>`

Daily builds(Electric Commander) + monitoring(CDASH)

- Automated building of the whole software system every day
  - Problems immediately visible
  - Several platforms are tested.

LCGSoft

Dashboard Calendar Previous Current Project

No file changed as of **Monday, March 31 2014 - 03:00 CEST** Show Filters Advanc

Experimental

Site	Build Name	Update	Configure		Build		Test		
		Files	Error	Warn	Error	Warn	Not Run	Fail	Pass
ec-slc6-x86-64-spi-8	x86_64-slc6-gcc48-opt	0	0	0	4^{+4}	1^{+1}	0	16^{+14}	72^{+1}_{-13}

Preview

Site	Build Name	Update	Configure		Build		Test		
		Files	Error	Warn	Error	Warn	Not Run	Fail	Pass
ec-slc6-x86-64-spi-9	x86_64-slc6-gcc48-opt	0	0	0	0	0	0	0_{-2}	88^{+2}
ec-slc6-i686-spi-1	i686-slc6-gcc48-opt	0	0	0	0	0	0	0_{-1}	69^{+1}
lxbsp0516.cern.ch	x86_64-slc5-gcc43-opt	0	0	0	0	0	0	0	69
ec-slc6-x86-64-spi-7	x86_64-slc6-icc14-dbg	0	0	0	27	1	0	38	31
macitois17.cern.ch	x86_64-mac108-gcc42-opt	0	0	0	10_{-6}	1	0	15_{-4}	32^{+4}

Release

Site	Build Name	Update	Configure		Build		Test		
		Files	Error	Warn	Error	Warn	Not Run	Fail	Pass
ec-slc6-x86-64-spi-8	x86_64-slc6-gcc48-opt	0	0	0	0	0	0	0	91^{+4}
lxbsp0516.cern.ch	x86_64-slc5-gcc43-opt	0	0	0	0	0	0	0_{-1}	71^{+1}
macitois17.cern.ch	x86_64-mac108-clang34-opt	0	0	0	19	1	0	7	41
macitois18.cern.ch	x86_64-mac108-gcc42-opt	0	0	0	21^{+2}	3^{+2}	0	7	41
ec-slc6-x86-64-spi-9	x86_64-slc6-gcc47-opt	0	0	0	0	0	0	0	71

Status and plans

- Almost all generators requested by ATLAS, CMS and LHCb implemented in the new system (this is “work in progress”)
- Since december the releases are done entirely using CMake machinery by a “single click”
- Ready to be used for automatic installation by ‘non-genser experts’
- Can provide “alpha” testing of new versions (existing example: release candidate testing for Rivet).
- Ideas about further development of the testing system from the generator authors are welcome.