# GENSER ACTIVITIES

Dmitri Konstantinov for the GENSER team

# Introduction

GENSER project has started more than 15 years and main activity was integration and deployment of new generators into AFS.

At that time there was no concept of LCG releases and “externals” were installed **directly** to AFS under “external” directory and all generators under ‘external/MCGenerators’ using ‘automake’-based system written by Oleg Zenin.

Now: GENSER2 -> LCGCMAKE developed by Pere

Team: Grigory Latyshev,  Ivan Razumov and me.

Organization: shifts covering entire year (~1 FTE)

All activities under SFT SPI umbrella.

# Difference between “externals” and “generators”

In LCGCMAKE there is no difference between “generators” and any other package.

```
#---mcutils-------------------------------------------
LCGPackage_Add(
  mcutils
  URL ${gen_url}/mcutils-<NATIVE_VERSION>.tar.gz
  CONFIGURE_COMMAND <VOID>
  BUILD_COMMAND <VOID>
  INSTALL_COMMAND make install PREFIX=<INSTALL_DIR>
  BUILD_IN_SOURCE 1
  DEPENDS heputils
)
```

```
LCG_external_package(pythia8          210           ${MCGENPATH}/pythia8 )
# Pythia 8 version for vincia 2.0.01
LCG_external_package(pythia8          223           ${MCGENPATH}/pythia8 )
LCG_external_package(pythia8          219           ${MCGENPATH}/pythia8 author=219 )
LCG_external_package(pythia8          219.pdf6plugin ${MCGENPATH}/pythia8 author=219 lhapdf6plugin=LHAPDF6.h )
LCG_external_package(pythia8          235           ${MCGENPATH}/pythia8 )
LCG_external_package(pythia8          230           ${MCGENPATH}/pythia8 )
```

But long time ago we agreed that:

- Allowed to have more than 1 version in LCG release
- Existing release can be updated with new generators
- Not desirable to have “external” depending on “generator”

NB: Different patches (settings) can be applied for different experiments (MadGraph, ATLAS)

# GENSER now

Our main responsibilities:

- Integration of new generators into LCGCMAKE (quite rare now)
- Patching generators to comply with the latest compilers (gcc7, gcc8)
- Following ATLAS and LHCb requests we update of existing LCG releases with new generators versions.

  It includes:
  - Update of very old LCG releases (i.e 67c) where installation is performed directly to AFS, no rpms, not taf-files
  - Update of LCG releases which use rpm installation
  - Update of LCG releases which use rmp installation and also installation from tar-files. (>= 84a)

- Sanity checks and deployments to AFS/CVMFS of LHAPDF sets provided by PDF community
- Development of auxiliary tools for LCGCMAKE and GENSER needs
  - LCGENV
  - LCGCMAKE post-install script
  - LCGVIEW
  - Python scripts managing installation from tar-balls.

# LHAPDF sets: sanity checks and deployment (https://lhapdf.hepforge.org/ , Andy Buckley and Co)

At the very beginning at LHAPDF5 era for each LHAPDF5 version we had corresponding LHAPDF set directory and it was very space consuming.

LHAPDF 5.8.1 and LHAPDFSETS/5.8.1

Starting with LHAPDF6 we have just “current” (or latest) PDF sets. Not part of LCG release.

Total disk space taken by all PDF sets ~ 62G

- AFS: /afs/cern.ch/sw/lcg/external/lhapdfsets/current/
- CVMFS: /cvmfs/sft.cern.ch/lcg/external/lhapdfsets/current/

Workflow:
1) Contacted by PDF authors with request to deploy new PDF set (1-2 times per month)
2) Run sanity checks
3) Deployed to AFS and CVMFS
4) Announced by LHAPDF authors with new LHAPDF release.

Git repo: https://gitlab.cern.ch/GENSER/lcgenv

# LCGENV – runtime environment for LCG packages

By default LCGCMAKE removes LD_LIBRARY_PATH from shared libraries, experiments use LCG release from their own frameworks which deals with environment variables for them.

Q: But how to use LCG release by usual user/physicist? How to set runtime environment properly?

A: LCGENV!

For this purpose we developed a dedicated Python script which reads in release metafiles and spits out environment for given package.

*Additional non-standard variables can be added directly to lcgenv package*

herwig++ 105 Bytes

```
export REPO=${HERWIGPP_HOME}/share/Herwig++/HerwigDefaults.rpo
export LHAPATH=$LHAPDF_HOME/share/PDFsets
```

export LCGENV_PATH=/afs/cern.ch/sw/lcg/releases

lcgenv -p LCG_79 x86_64-slc6-gcc48-opt Pythia8 219

N.B. each package contain bash script which uses LCGENV to set environment

pythia8env-genser.sh, i.e. <package_name>pythia8env-genser.sh

Git repo: https://gitlab.cern.ch/GENSER/lcgenv

# post-install.sh (part of LCGCMAKE) – solution for relocation problem?

Preparing LCG release initially we install it to temporary directory and then it is packed to RPM or/and to tar-file. After that using RPM/tar installation system we deploy it to AFS/CVMFS.

Q: “De facto” we move library/package from initial installation path, i.e we relocate package. Sometime it can cause problem at runtime. How to solve this problem at least partially?

A: Running LCG build we check all ascii files for hard-coded paths and save info about such files and hardcoded paths into auxiliary hidden file .filelist

For example,

/cvmfs/sft.cern.ch/lcg/releases/LCG_93/MCGenerators/pythia8/235/x86_64-slc6-gcc7-opt/.filelist

/build/jenkins/workspace/lcg_release_tar/BUILDTYPE/Release/COMPILER/gcc7binutils/LABEL/slc6/install/MCGenerators/lhapdf/6.2.0/x86_64-slc6-gcc7-opt->MCGenerators/lhapdf/6.2.0-46baa/x86_64-slc6-gcc7-opt

```
bin/pythia8-config
share/Pythia8/examples/Makefile.inc
```

at “post-install” step of rpm or tar-based installation systems we use this information and replace “hard-coded” paths.

# GENSER plans for 2018

- Continue routine duties listed in previous slides.
- We are working on the following items:
  - Regular static code analysis using “coverity” instance (coverity.cern.ch) managed by SFT group for popular MC generators and tools, such as Pythia8, LHAPDF, Rivet
  - Development of infrastructure allowing regression testing and physics validation of basic observable
- Further participation in SPI activities!