Configuring external libraries to CMSSW (like Geant4)

How does it work ?

After setting up CMSSW project workspace (`scram project ...`) following directory structure will appear:

```
bin  config  doc  external  include  lib  logs  python  share  src  test  tmp
```

Our own modules and plugins with code stored in "src" will be compiled into `lib/slc4_ia32_gcc345/` directory, to the files `lib*.so`, `plugin*.so` and `*.edmplugin`. Those files are mainly shared libraries, which depends on external libraries.

External libraries are installed in the directory `/afs/cern.ch/cms/sw/slc4_ia32_gcc345/external/` (for example official Geant4 libraries are here: `/afs/cern.ch/cms/sw/slc4_ia32_gcc345/external/geant4/9.2/lib/`). In case of local CMSSW installation, local directory shall be used instead of `/afs/cern.ch/cms/sw/`.

As there are many locations of different libraries there is a system of XML configuration files, where you can find necessary information (i.e. library locations) for all external tools/libraries. You can find it in the local project workspace, under `config/toolbox/slc4_ia32_gcc345/tools`.

After changing something in the XML files one needs to cofigure CMSSW workspace properly. This can be done with scram. Useful commands:

`scram tool remove geant4` - tool removal

`scram setup geant4core` - tool installation (configuration)

`scram tool info geant4` - prints information about tool configuration

Moreover `ldd` can give some important hints on library dependencies.

Step by step instruction how to replace Geant4 by custom instance

- Source default set of environment variables:

```
> source /afs/cern.ch/cms/sw/cmsset_default.sh
```

- Initialize CMSSW project space:

```
> scram project CMSSW CMSSW_3_1_1
```

- Replace Geant4 configuration by custom one, pointing to Geant4 compiled in `/afs/cern.ch/exp/totem/scratch/Release/tests/custom_g4` directory:

```
> cd CMSSW_3_1_1
> find . -name "geant4.xml"
./config/toolbox/slc4_ia32_gcc345/tools/available/geant4.xml
./config/toolbox/slc4_ia32_gcc345/tools/selected/geant4.xml
> cp /afs/cern.ch/exp/totem/scratch/Release/tests/custom_g4/geant4.xml ./config/toolbox/slc4_ia32
> cp /afs/cern.ch/exp/totem/scratch/Release/tests/custom_g4/geant4.xml ./config/toolbox/slc4_ia32
> find . -name "geant4core.xml"
./config/toolbox/slc4_ia32_gcc345/tools/available/geant4core.xml
./config/toolbox/slc4_ia32_gcc345/tools/selected/geant4core.xml
> cp /afs/cern.ch/exp/totem/scratch/Release/tests/custom_g4/geant4core.xml ./config/toolbox/slc4_
> cp /afs/cern.ch/exp/totem/scratch/Release/tests/custom_g4/geant4core.xml ./config/toolbox/slc4_
```

- Reconfigure Geant4 tool, by typing:

```
> scram setup geant4
> scram setup geant4core
```

- Check configuration:

```
> scram tool info geant4
(...)
GEANT4_BASE=/afs/cern.ch/exp/totem/scratch/Release/tests/custom_g4
(...)
> scram tool info geant4core
(...)
GEANT4_BASE=/afs/cern.ch/exp/totem/scratch/Release/tests/custom_g4
(...)
```

- Now you can fetch source code from SVN repository and compile it

- Check if SimG4Core/Application module was linked with custom G4 libraries:

```
> ldd lib/slc4_ia32_gcc345/libSimG4CoreApplication.so | grep G4process
   libG4processes.so => /afs/cern.ch/exp/totem/scratch/Release/validation/rel3.3.tests/CMSSW_3_1_
> ls -l /afs/cern.ch/exp/totem/scratch/Release/validation/rel3.3.tests/CMSSW_3_1_1/external/slc4_
lrwxr-xr-x  1 lgrzanka zj 86 Apr 22 15:15 /afs/cern.ch/exp/totem/scratch/Release/validation/rel3.
```

- In case of problems repeat following steps:

```
> scram tool remove geant4
> scram tool remove geant4core
> scram setup geant4
> scram setup geant4core
> scram b -j 4
> scram b -j 4
> scram b
```

Geant4 compilation

Sources of Geant4 libraries used in CMSSW are provided in the `/afs/cern.ch/cms/sw/slc4_ia32_gcc345/external/geant4/9.2.p01`. Let us copy it into some directory and go there.

First thing is to adjust `etc/G4BuildConf.sh` which contains all enviromental variables used in compilation process. In the original file lot of variables pointed to strange `/build/d94/BUILD/...` directory, probably on one of CMS build machines. Our file will have all the variables pointing to some subdirectory of `/afs/cern.ch/exp/totem/scratch/Release/tests/custom_g4` directory (`G4TOTEM` variable), see file `/afs/cern.ch/exp/totem/scratch/Release/tests/custom_g4/etc/G4BuildConf.sh`.

Second thing is the compiler. Geant4 needs to be compiled with some old GCC (3.3.?). In order to use the same compiler as in CMSSW, following enviromental variables are overwritten in `G4BuildConf.sh` file:

```
export BASE=/afs/cern.ch/exp/totem/scratch/Release/cmssw
export GCC_BASE=$BASE/slc4_ia32_gcc345/external/gcc/3.4.5-cms
export GCCBINDIR=$GCC_BASE/bin
export CXX=$GCCBINDIR/c++
export LD_LIBRARY_PATH=$GCC_BASE/lib
export PATH=$GCCBINDIR:$PATH
export MAKE_BASE=$BASE/slc4_ia32_gcc345/external/gmake/3.81-cms2
export PATH=$MAKE_BASE/bin:$PATH
```

We can now start compilation. First we need a fresh terminal ! Remember to compile Geant4 on a console which was not used before to compile CMSSW. Let us now go to `source` subdirectory and type:

```
source ../etc/G4BuildConf.sh
make
make global
```

We should now expect some files created in `../lib/Linux-g++`. This is slightly different than original location (`../lib`), but it is easy to adjust it in `geant4.xml` and `geant4core.xml`.

Finally if we want to compile Geant4 with debugging options, the file to inspect is `../etc/architecture.gmk`. One should put there somewhere at the beginning:

```
G4DEBUG := 0
G4NO_OPTIMISE := 0
G4OPTIMISE := 1
```

-- LeszekGrzanka - 22-Apr-2010

This topic: TOTEM > CompOfflineTools
Topic revision: r4 - 2010-06-21 - LeszekGrzanka