

# Table of Contents

<b>Offline Simulation Software CMSSW_6_2_0 - getting started.....</b>	<b>1</b>
Requirements.....	1
For experts.....	1
Documentation.....	1
STEP 0.....	1
STEP 1. Getting source.....	1
STEP 2. Building.....	2
STEP 3. Running example.....	2
STEP 4. Generating plots.....	3

# Offline Simulation Software CMSSW\_6\_2\_0 - getting started

This document will explain how to use TOTEM offline software.

## Requirements

- SLC6 Linux 64bit with afs access. Can be lxplus.
- Totem svn access.
- 9GB Free disk space.

## For experts

If you know what to do, use following instruction (if not, read whole page):

```
export SCRAM_ARCH=slc6_amd64_gcc472
source /afs/cern.ch/cms/cmsset_default.sh
scram project CMSSW CMSSW_6_2_0
svn co svn+ssh://svn.cern.ch/repos/totem/branches/CMSSW_6_2_0/offline/cmssw/src/ CMSSW_6_2_0/src/
cd CMSSW_6_2_0
scram build -j 15
```

If you want to use cmsRun to run cmssw configuration files type:

```
cmsenv
```

in CMSSW\_6\_2\_0 directory.

## Documentation

- The CMS Offline WorkBook, especially:
  - ◆ Computing Environment
  - ◆ About Scram
  - ◆ CMSSW Configuration Files

## STEP 0.

Let us login via ssh to some remote machine with SLC6 operating system. It might be lxplus.

```
ssh -X psikora@lxplus
```

## STEP 1. Getting source

Let us create some temporary directory

```
mkdir tmp
cd tmp
```

We have to load default set of environment variables which are necessary for work with CMSSW. Script containing these variables is stored in the directory where CMSSW framework and all necessary libraries are installed.

Machine	Directory	Comments
lxplus, lxbatch, machines with AFS access	/afs/cern.ch/cms/	CMS installation

Let us use default CMS installation, type in bash:

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

Now we have access to `scram` command. We can check that by typing:

```
scram help
```

## STEP 2. Building

We will compile our software from sources.

First we initialize CMSSW project area:

```
scram project CMSSW CMSSW_6_2_0
```

After that your workspace called **CMSSW\_6\_2\_0** is created, this is the default name for the working directory. It is also possible to specify another directory as your workspace by:

```
scram project -n <your_given_name> CMSSW CMSSW_6_2_0
```

We can see that after that command following directory structure emerged:

```
[psikora@lxplus0389 tmp]$ ls CMSSW_6_2_0/
bin  cfipython  config  doc  include  lib  logs  objs  python  src  test  tmp
```

Our sources shall go to `CMSSW_6_2_0/src` subdirectory, so let us first check out the code from Totem SVN to that directory. If you have source code already stored in another place, do not try to make any links from `CMSSW_6_2_0/src` there, as some strange unexpected problems might appear.

```
[psikora@lxplus0440 tmp]$ svn co svn+ssh://svn.cern.ch/repos/totem/branches/CMSSW_6_2_0/offline/cm
[psikora@lxplus0440 tmp]$ ls CMSSW_6_2_0/src
Configuration      LlTriggerTotem   SimG4Core         TotemRawData
DataFormats        RecoTotemRP      SimTotem          TotemRPValidation
GeneratorInterface RecoTotemTlT2    TotemAlignment    TotemTlT2Validation
Geometry           SimDataFormats  TotemAnalysis
IOMC               SimG4CMS         TotemCondFormats
```

Now let us start compilation. Using 15 threads (`-j 15`) it takes 10-15 minutes.

```
cd CMSSW_6_2_0/
scram build (or scram b)
[ scram b -j 15 is another option to run build which will use 15 threads to compile the code. ]
```

## STEP 3. Running example

Firstly, set environment variables to use your CMSSW installation and get access to `cmsRun` script. This is done by calling `cmsenv` in workspace directory.

```
[psikora@lxplus0389 CMSSW_6_2_0]$ cmsenv
```

Some example configuration files are in

`CMSSW_6_2_0/src/Configuration/TotemStandardSequences/test/` Let us take example configuration file:

## CompOfflineGettingStartedCMSSW620 < TOTEM < TWiki

```
CMSSW_6_2_0/src/Configuration/TotemStandardSequences/test/RP/prodRPelasticBeta90Energy7TeV_cfg.py
```

Check number of events to generate:

```
process.maxEvents = cms.untracked.PSet(
  input = cms.untracked.int32(10)
)
```

Check output file:

```
exec 'process.' + str(process.outpath) + '.fileName = cms.untracked.string("file:prodRPelasticBeta90Energy7TeV.root")
```

Check sequence of modules to execute:

```
process.p1 = cms.Path(process.generator*
process.SmearingGenerator*
process.OptInfo*
process.g4SimHits*
process.mix*
process.RPSiDetDigitizer*
process.RPClustProd*
process.RPHecoHitProd*
process.RPSinglTrackCandFind*
process.RPSingleTrackCandCollFit*
process.ElasticReconstruction*
process.RPCC)
```

Now we can run that file:

```
[psikora@lxplus0389 CMSSW_6_2_0]$ cmsRun src/Configuration/TotemStandardSequences/test/RP/prodRPelasticBeta90Energy7TeV_cfg.py
```

If everything went OK, then we can find output file:

```
[psikora@lxplus0389 CMSSW_6_2_0]$ ls -l *root
-rw-r--r--. 1 psikora zj 124956 Sep 19 10:41 prodRPelasticBeta90Energy7TeV.root
```

## STEP 4. Generating plots

Now we can generate some plots.

We can start root session, to see plots:

```
[psikora@lxplus0389 CMSSW_6_2_0]$ root -l
root [0] TBrowse t
```

\_Pawel Sikora - 19-Sep-2013\_

---

This topic: TOTEM > CompOfflineGettingStartedCMSSW620

Topic revision: r3 - 2014-06-17 - LeszekGrzanka



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback