# Table of Contents

# Example Software Chain

## Detector Geometry

The detector geometry is described in the so called compact XML format (see CompactXml). The compact file is then translated into the different detector descriptions required by other applications using GeomConverter.

A large number of detector models is available from http://lcsim.org/detectors/⧉. Each zip file contains the compact description as well as the translated file formats required for this example. It also contains conditions data like calorimeter calibration values.

## Detector Simulation

The Geant4 application SLIC is used for the full detector simulation. It requires a detector geometry description in the lcdd format and accepts generator files in the StdHep and the LCIO file format. Define an alias to run SLIC (no additional environment variables required):

```
alias slic="/afs/cern.ch/eng/clic/software/slic/2.9.8/rhel5_i686_gcc/scripts/slic.sh"
```

Download and unzip the detector description for CLIC_SID_CDR:
http://www.lcsim.org/detectors/clic_sid_cdr.zip⧉. For the input events you can use this example file:
bbudsc_3evt.stdhep.

Run the example and create `bbudsc_3evt_SIM.slcio` which contains the simulated hits for the 3 input events:

```
slic -i bbudsc_3evt.stdhep -g clic_sid_cdr.lcdd -o bbudsc_3evt_SIM.slcio -r 3
```

Usually the Monte Carlo events are generated as head on collisions while the machine is in fact operated under a small crossing angle. The correct angular distribution can be obtained by applying a Lorentz transformation to all four vectors of half the crossing angle. Download the macro defaultClicCrossingAngle.mac and use it in SLIC:

```
slic -i bbudsc_3evt.stdhep -g clic_sid_cdr.lcdd -o bbudsc_3evt_SIM.slcio -r 3 -m defaultClicCross
```

💡 If you run SLIC more than once you will have to delete the output file beforehand or add the `-x` flag, since SLIC does not overwrite files by default.

## Event Reconstruction

The full event reconstruction is performed in three steps using LCSim and SlicPandora.

### Tracker Hit Digitization and Track Reconstruction

Define alias to use the CLIC CDR version of lcsim:

```
alias lcsim="java -jar /afs/cern.ch/eng/clic/software/lcsim/CLIC_CDR/target/lcsim-CLIC_CDR-bin.ja
```

Download the steering file clic_cdr_prePandora.lcsim and the tracking strategies defaultStrategies_clic_sid_cdr.xml and run LCSim:

```
lcsim clic_cdr_prePandora.lcsim -DinputFile=bbudsc_3evt_SIM.slcio -DoutputFile=bbudsc_3evt_TRACKI
```

💡 The steering file can contain variables defined by `${variableName}` which have to be set at execution time by adding `-DvariableName=value` to the command line parameters.

💡 LCSim reads the detector geometry directly from the compact XML which is picked up from http://www.lcsim.org/detectors ⧉ based on the detector name set in the input LCIO file.

## Calorimeter Clustering and Particle Flow Reconstruction

Setup the environment:

```
source /afs/cern.ch/eng/clic/software/x86_64-slc5-gcc41/SIDSOFT/CLIC_CDR/slicPandora_env.sh
```

Download the steering file PandoraSettings_CLIC_CDR.xml and run SlicPandora using the geometry file from http://www.lcsim.org/detectors/clic_sid_cdr.zip ⧉:

```
PandoraFrontend clic_sid_cdr_pandora.xml PandoraSettings_CLIC_CDR.xml bbudsc_3evt_TRACKING.slcio
```

## Truth Linking and DST Output:

The last step is optional. It adds LCRelations between the MCParticles and the reconstructed particles, clusters and tracks to persist the truth links without the information of individual hits. Two output files are created: the DST file only contains the original four vectors (MCParticles) and the high level reconstructed objects, while the REC file has the full information including all hits.

Download the steering file clic_cdr_postPandora.lcsim and run LCSim:

```
lcsim clic_cdr_postPandora.lcsim -DinputFile=bbudsc_3evt_PFA.slcio -DrecFile=bbudsc_3evt_REC.slci
```

This topic: CLIC > SiDSoftwareFullExample
Topic revision: r2 - 2013-07-14 - ChristianGrefe