

Defining a Detector

A number of detectors have already been defined. To define your own, it is best to start with one of them and modify as necessary.

Full Detector Description

The detector geometry is defined using the compact format. Instructions for creating this detector description file can be found at [Creating a New Detector Description](#).

The resulting lcmd file needs to be checked for overlaps. The easiest way to do this is to create a macro file containing the following command

```
/geometry/test/recursive_test .
```

To check for overlaps:

```
> slic -g detector.lcmd -z -m geom_chk.mac
```

There may be a few spurious overlaps within calorimeters due to roundoff errors. All other overlaps should be resolved.

Fast Monte Carlo Files

The current fast Monte Carlo smears charged tracks covariantly, and creates clusters for neutral particles, treating photons and neutral hadrons separately. The fast Monte Carlo requires the following properties files to provide input:

TrackParameters.properties

The minimum transverse momentum as well as angular fiducial regions are defined here. The names of the track smearing files are also listed here. These files are generated using the `lcmdtrk` program. These files should reside in the directory `TrackParameters`. The file `SimpleTrack` provides the ability to define momentum smearing parameters independent of the `lcmdtrk` parameters.

ClusterParameters.properties

This file contains the fiducial cuts on momentum and angular coverage as well as expected energy resolution terms for the electromagnetic and hadronic calorimeters. Electromagnetic and hadronic calorimeter resolution terms will be scaled to achieve the jet energy resolution specified by parameter `JETResolution` if the boolean parameter `JETParameterization` is set to true. To perform this scaling properly you must also specify the fraction of the jet energy that is electromagnetic (`JETEMEnergyFraction`) and the fraction due to neutral hadrons (`JETHadEnergyFraction`). You can choose to only scale the hadronic energy resolution (`JETHadDegradeFraction=1`), to only scale the electromagnetic resolution (`JETHadDegradeFraction=0`), or to scale both using an intermediate value for `JETHadDegradeFraction`. More information on the jet energy resolution algorithm can be found on pp. 162-163 of the [SiD Detector Outline Document](#).

IDEfficiency.properties

This provides a simple method for simulating the confusion terms in identifying individual particles. In addition, the parameter `wt_charged_track_calorimeter_energy` determines how often a charged hadron's reconstructed energy will be set equal to the calorimeter energy. For example, if `wt_charged_track_calorimeter_energy=0` then the tracker momentum will always be used for the charged hadron's reconstructed energy; if `wt_charged_track_calorimeter_energy=1` then the calorimeter energy will always be used for the charged hadron's reconstructed energy.

SimpleTrack.properties

This file contains the parameters for smearing the transverse momentum p of charged tracks according to the simple formula $p/p^2 = a + b/\psi$. This method of smearing track parameters is enabled by setting the `MCFast` class constructor argument `simple=true`. The file parameter `ConstantTerm = a` and the file parameter `ThetaTerm = b`. By default, the errors for the track angles and impact parameters are taken from the files defined in `TrackParameters.properties`. However, the angle and impact parameter errors can be scaled relative to these default errors using the `SimpleTrack.properties` parameters `TanLambdaErrorScale`, `PhiErrorScale`, `D0ErrorScale` and `Z0ErrorScale`.

Reconstruction Files

Digisim

A `digisim.steer` file is needed in the `digisim` directory. This file contains timing and threshold cuts for the hits in the calorimeters. Using the same file as a previous detector is fine, with the following caveats: Scintillator and gas calorimeters have different hit thresholds, so use the appropriate file. Also, files for detectors prior to `sid01` do not contain muon calorimeter information.

Calorimeter Sampling Fractions

(see also the [lcsim-cal-calib](#) package)

Sampling fractions need to be provided for each calorimeter defined in the detector. Currently there needs to be one file for each calorimeter in the directory `SamplingFractions`. The files may contain different values for different layers or ranges of layers (needed for `Ecal`), and a digital flag for the hits to be treated digitally. These numbers are used to give an energy estimate of hits and clusters without angle or particle type corrections.

Generating the numbers

Once the `digisim.steer` file is in place, a driver `QSFCalibrationFromData` (soon to be in CVS) can be run to generate the numbers for each calorimeter (or sub-calorimeter). The procedure minimizes dE/\sqrt{E} per event, using only the calorimeter information. Three sets of numbers are generated. (`Ecal+Hcal`, `Ecal+Hcal+Muon`, `Ecal+Hcal+MuonEndcap`) The input files must be the qq events at fixed energies. The numbers will vary significantly depending on the cm energy, so typically the driver is run over a combination of 10k events each at 100,200,500 GeV cm energy. The numbers can then be placed in the Sampling fraction files.

Calibration files

The calibration files reside in the `photonCalibration` and `hadronCalibration` directories. The constants are derived by a driver (`QCalibrationFromData`, soon to be in CVS) which uses the cheat reconstruction. Therefore, in addition to the sampling fraction files, all the `FastMC` files must also exist. The procedure is to attach calorimeter hits to the neutral particles, calculate sampling fractions independently for photons and neutral hadrons minimizing dE/\sqrt{E} assuming an angle correction, and calculate response corrections (for nonlinearity). The calibration depend on the distributions in energy, angle, and particle type, and is typically run on the ZZ events at 500 GeV. The files `photon-(slic version).properties` and `nh-(slic version).properties` can then be created, and are used by the classes `QPhotonClusterEnergyCalculator` and `QNeutralHadronClusterEnergyCalculator`.

LongitudinalHMatrix

To run the pflow reconstructions, the file `LongitudinalHMatrix.hmx` is needed. Ask Norman Graf.