## Subject: CbmGenerators and MCEvent
## Posted by Volker Friese on Tue, 26 May 2009 08:55:20 GMT

View Forum Message <> Reply to Message

Latest chages in the framework offer the possibility to define experiment-specific classes for event information on the MC level.

For CBM, there is now the class CbmMCEvent (library cbmdata). It is very similar to the (still existing) class FairMCEventHeader, but adds the variable event plane angle. Further additions can be made on request.

The new class will be in the branch MCEvent. The old branch MCEventHeader (class FairMCEventHeader) is still present such as not to interfere with other fairroot users.

In the simulation macros, the instantiation of FairPrimaryGenerator must be replaced by the one of CbmPrimaryGenerator:

```
CbmPrimaryGenerator* primGen = new CbmPrimaryGenerator();
fRun->SetGenerator(primGen);
```

The proper filling of the CbmMCEvent class is in the responsibility of the concrete generator class. The PrimaryGenerator takes care for the event vertex, event time and number of primary tracks. All other variables have to be filled by the concrete classes (derived from FairGenerator). Example:

```
// -----   Public method ReadEvent   -------------------------------------
Bool_t CbmHsdLvmGenerator::ReadEvent(FairPrimaryGenerator* primGen) {

  ....
  ....

  // --> Convert to CbmPrimaryGenerator
  CbmPrimaryGenerator* cbmGen = (CbmPrimaryGenerator*) primGen;

  // Set event information
  CbmMCEvent* event = cbmGen->GetEvent();
  if ( ! event->IsSet() ) {
    event->SetEventID(++fCurrentEvent);
    event->SetB(b);
    event->SetPhi(0.);
  }

}
```

## Subject: CbmHsdLvmGenerator
## Posted by Volker Friese on Tue, 26 May 2009 09:05:59 GMT

View Forum Message <> Reply to Message

On special request, there is now the new generator class CbmHsdLvmGenerator. It reads a special output file of HSD containing vector mesons (one per event), decays each meson either into electron or muon pairs and adds the daughters to the stack.

The need for this generator came up since the decay with PYTHIA did not take into account the proper mass of the vector mesons (it takes the PDG mass, not the (e.g. in-medium modified) one provided by the model).

Usage:
CbmHsdLvmGenerator* gen = new CbmHsdLvmGenerator(inFile, pBeam, iMode);
where inFile is the name of the input file, pBeam the beam momentum in GeV per nucleon and iMode the decay mode (1 = e+e-; 2 = mu+mu-).

The format of the input file is: One line per event / meson containing
m  px  py  pz  E  t1  t2  b
If |t2/t1]>0.001, the event is ignored (speciality of HSD).

---

## Subject: CbmUrqmdGenerator
## Posted by Volker Friese on Tue, 26 May 2009 10:58:24 GMT

View Forum Message <> Reply to Message

The interface to the UrQMD output file ftn 14 has been extended to

automatically read the lab transformation beta from the UrQMD event header file and perform the appropriate Lorentz transformation to the lab system. It is thus not required any longer that the UrQMD file be produced in the CM system.

allow for an event plane angle different from zero, a functionality not provided by UrQMD itself. The user can call the method CbmUrqmdGenerator::SetEventPlane(phiMin, phiMax); then, a random event plane angle is produced with a flat distribution between phiMin and phiMax (to be specified in radians). All track momenta are rotated accordingly before being passed to the stack.

store the event plane angle in the CbmMCEvent class (branch MCEvent).

This new functionality is provided by the class CbmUrqmdGenerator (library cbmgenerators). Of course, you are free to use FairUrqmdGenerator if you do not need this functionality. However, then the impact parameter b will not be stored in CbmMCEvent (but, as before, in FairMCEventHeader).