

Hi!,

This problem is now solved and the corrected code is available in trunk.
In the original code the following was declared and used without proper initialization:

```
map<Int_t, Float_t> fTrackEnergy;  
map<Int_t, Float_t> fTrackTime; //time of first point  
map<Int_t, std::vector <Int_t> > fTrackMcTruth; //McTruth  
fTrackEnergy.clear();  
fTrackTime.clear();  
fTrackMcTruth.clear();  
.....  
.....  
point = (PndEmcPoint*) fPointArray->At(iPoint);  
fTrackEnergy[point->GetDetectorID()] += point->GetEnergyLoss();  
point_time=point ->GetTime();  
if (point_time < fTrackTime[point->GetDetectorID()])  
    fTrackTime[point->GetDetectorID()] = point_time;
```

In the lines above we can see a comparison and a "+" operation on not initialized member of the map which has(might have) an undefined state.

Another point, which is (in this case) a matter of taste and beauty, is the declaration of:

```
PndEmcPoint* point = NULL;
```

```
map<Int_t, Float_t>::const_iterator p;
```

outside the loop. In this case they are not leading into wrong computations but potentially they can, as their values are changed inside the loop.

Greets,
/M
