
Subject: Re: Specific event selection in Dpm generator
Posted by [donghee](#) on Thu, 09 Feb 2012 10:27:25 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello everyone,

I solve this problem with quite simple trick using while loop only in PndDpmDirect code.(posted below)

It might be useful that someone prepare a selector function inside PndDpmDirect in the future, if it is not exsiting.

Anyhow, thank you again for your reading...

Quote:

```
// ----- Public method ReadEvent -----
```

```
Bool_t PndDpmDirect::ReadEvent(FairPrimaryGenerator* primGen) {
```

```
    int npart, i;
    Double_t fX, fY, fZ, radius;
    double Px[1000],Py[1000],Pz[1000],E[1000],Pm[1000],Wh[1000];
    int Id[1000];
```

```
    float Generator=0.; // Format in which events are produced (0=pythia, 1=pluto)
```

```
    Double_t weight = 1.0;
    Int_t activeCnt=0;
```

```
    Int_t NIDs=0;
    Int_t event_type = 844;
    Int_t particle_number = 4;
    double PID_type_1 = 211;
    double PID_type_2 = -211;
```

```
    //4pi = 844
    //2pi = 422
```

```
    ////////////////////////////////////////Donghee Kang
    //Selector!!!
```

```
    while(1){
```

```
        bool selector = false;
        // run generator
        dpm_gen__(&Generator, &fSeed);
```

```
        // Loop over all produced particles
        npart = lujets_.n;
```

```
        if(npart == particle_number){//////////////////////////////////Donghee Kang
            NIDs = 0;
            for (i= 0; i< npart; ++i) {
                Id[i]=lujets_.k[i+1000];
```

```

    if( (Id[i] == PID_type_1) || (Id[i] == PID_type_2) ){
        NIDs += TMath::Abs(Id[i]);
    } //Id[i] check!!!
} //loop
if(NIDs == event_type){
    for (i= 0; i< npart; ++i) {
        Id[i]=lujets_.k[i+1000];
        Px[i]=lujets_.p[i];
        Py[i]=lujets_.p[i+1000];
        Pz[i]=lujets_.p[i+2000];
        Pm[i]=lujets_.p[i+4000];
        E[i]=lujets_.p[i+3000];
        Wh[i]=1.0;

        /* Check if fGasmode is set */
        fX = 0.;
        fY = 0.;
        fZ = 0.;
        if (fGasmode == 1) {

            // define position of track start
            // Random 2D point in a circle of radius r (simple beamprofile)
            radius = gRandom->Gaus(0,fRsigma);
            gRandom->Circle(fX, fY, radius);

            // calculate fZ according to some (probability) density function of the gas
            fZ=fDensityFunction->GetRandom();

        }

        // add track
        //printf("- I -: new particle at: %f, %f, %f ...\n", fX, fY, fZ);
        primGen->AddTrack(Id[i], Px[i], Py[i], Pz[i], fX, fY, fZ);
    } //loop
    selector= true;

} ///////////////////////////////////////////////////
else{
    selector = false;
}
} //npart == 4
if(selector){
    //cout<<"Yes I have 4 pi+"<<endl;
    break;
}
} //End of while
return kTRUE; ///////////////////////////////////

}

```

