
Subject: PandaRoot volumes for forward spectrometer
Posted by [Olaf Hartmann](#) on Wed, 21 Dec 2011 14:28:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear all,

I'm looking for the (most detailed available) geometry descriptions in PandaRoot for the forward detectors. What I can find in the geometry folder is:

dch.geo/dch.root
dipole.geo
fsc.geo
ftofwall.root
fts.geo
muon_FS.root

Can someone of the detector groups comment which geometries are up to date and how to initialize them from the macro?

In general I think it would be a good idea to maintain a WIKI page where the different geometries for the detectors are explained and which moreover is updated as soon as someone implements something new.

Cheers
Olaf.

Subject: Re: PandaRoot volumes for forward spectrometer
Posted by [Stefano Spataro](#) on Wed, 21 Dec 2011 18:10:24 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Olaf,
you can use the standard macro/pid/run_sim_sttcombi_pgun.C simulation macro.

If you want to have also forward Mdt, take out the "fast" part and put:

```
PndMdt *Muo = new PndMdt("MDT",kTRUE);  
Muo->SetMdtMagnet(kTRUE);  
Muo->SetBarrel("muon_TS_barrel_strip_v1_noGeo.root");  
Muo->SetEndcap("muon_TS_endcap_strip_v1_noGeo.root");  
Muo->SetForward("muon_Forward_strip_v1_noGeo.root");  
Muo->SetMuonFilter("muon_MuonFilter_strip_v1_noGeo.root");  
fRun->AddModule(Muo)
```

If you want to have the still untested bent pipe, uncomment the Pipe->SetGeometryFileName() line.

Hi Olaf,

Unfortunately, there is no usable macro for forward part inside repository.

I use the following geometry (part of the sim macro):

```
//----- CAVE -----  
  
FairModule *Cave= new PndCave("CAVE");  
Cave->SetGeometryFileName("pndcave.geo");  
fRun->AddModule(Cave);  
//----- Magnet -----  
FairModule *Magnet= new PndMagnet("MAGNET");  
//Magnet->SetGeometryFileName("FullSolenoid_V842.root");  
Magnet->SetGeometryFileName("FullSuperconductingSolenoid_v831.root");  
fRun->AddModule(Magnet);  
FairModule *Dipole= new PndMagnet("MAGNET");  
Dipole->SetGeometryFileName("dipole.geo");  
fRun->AddModule(Dipole);  
// ----- Pipe -----  
FairModule *Pipe= new PndPipe("PIPE");  
Pipe->SetGeometryFileName("beampipe_201112.root");  
fRun->AddModule(Pipe);  
//----- STT -----  
FairDetector *Stt= new PndStt("STT", kTRUE);  
Stt->SetGeometryFileName("straws_skewed_blocks_35cm_pipe.geo");  
fRun->AddModule(Stt);  
//----- MVD -----  
FairDetector *Mvd = new PndMvdDetector("MVD", kTRUE);  
// Mvd->SetGeometryFileName("Mvd-2.1_FullVersion.root");  
Mvd->SetGeometryFileName("Mvd-2.1_AddDisks_FullVersion.root");  
fRun->AddModule(Mvd);  
//----- GEM -----  
FairDetector *Gem = new PndGemDetector("GEM", kTRUE);  
// Gem->SetGeometryFileName("gem_3Stations.root");  
Gem->SetGeometryFileName("gem_4Stations.root");  
fRun->AddModule(Gem);  
//----- EMC -----  
PndEmc *Emc = new PndEmc("EMC",kTRUE);  
Emc->SetGeometryVersion(1);  
Emc->SetStorageOfData(kFALSE);  
fRun->AddModule(Emc);  
//----- DRC -----  
PndDrc *Drc = new PndDrc("DIRC", kTRUE);  
Drc->SetGeometryFileName("dirc_l0_p0.root");  
Drc->SetRunCherenkov(kFALSE);  
fRun->AddModule(Drc);  
//----- DISC -----
```

```

PndDsk* Dsk = new PndDsk("DSK", kTRUE);
Dsk->SetGeometryFileName("dsk.root");
Dsk->SetStoreCerenkovs(kFALSE);
Dsk->SetStoreTrackPoints(kFALSE);
fRun->AddModule(Dsk);
//----- MDT -----
PndMdt *Muo = new PndMdt("MDT",kTRUE);
Muo->SetBarrel("fast");
Muo->SetEndcap("fast");
Muo->SetMuonFilter("fast");
Muo->SetMdtMagnet(kTRUE);
Muo->SetMdtMFIron(kTRUE);
fRun->AddModule(Muo);
//----- FTS -----
FairDetector *Fts= new PndFts("FTS", kTRUE);
Fts->SetGeometryFileName("fts.geo");
fRun->AddModule(Fts);
//----- FTOF -----
FairDetector *FTof = new PndFtof("FTOF",kTRUE);
FTof->SetGeometryFileName("ftofwall.root");
fRun->AddModule(FTof);
=====

```

To have forward MDT use Sefano's suggestion.

Note, that default ftofwall.root from the svn might be overlapped with FSC. So I use my own ftofwall.root with shifted FTOF along z. You can create it using macro "macro/ftof/create_ftof_rootgeo.C"

Hope this helps.

Dmitry