

Creating / accessing detector geometry

Thursday, July 30
2

1st R3BRoot Development Workshop
July 28 - 30, 2015
GSI, Darmstadt



- Structure of a macro for ROOT geometry creation
- Accessing geometry parameters in the analysis



Create geometry file

- Geometry in TGeo format (ROOT)
- Macros for all detectors are in: `r3broot/macros/r3b/geo/*.C`
- Generated files are in `r3broot/geometry/*.root`



- Documentation:
- <https://root.cern.ch/drupal/content/root-users-guide-534>
- chapter 18 - “Geometry”



Structure of the macro

```
void create_tof_geo(const char* geoTag)
{
    // ----- Load media from media file -----
    FairGeoLoader*    geoLoad = new FairGeoLoader("TGeo","FairGeoLoader");
    FairGeoInterface* geoFace = geoLoad->getGeoInterface();
    TString geoPath = gSystem->Getenv("VMCWORKDIR");
    TString medFile = geoPath + "/geometry/media_r3b.geo";
    geoFace->setMediaFile(medFile);
    geoFace->readMedia();
    gGeoMan = gGeoManager;
    // -----

    // ----- Geometry file name (output) -----
    TString geoFileName = geoPath + "/geometry/tof_";
    geoFileName = geoFileName + geoTag + ".geo.root";
    // -----
}
```



Create materials

```
// ----- Get and create the required media -----  
FairGeoMedia*   geoMedia = geoFace->getMedia();  
FairGeoBuilder* geoBuild = geoLoad->getGeoBuilder();  
  
FairGeoMedium* mAir      = geoMedia->getMedium("Air");  
if ( ! mAir ) Fatal("Main", "FairMedium Air not found");  
geoBuild->createMedium(mAir);  
TGeoMedium* pMed2 = gGeoMan->GetMedium("Air");  
if ( ! pMed2 ) Fatal("Main", "Medium Air not found");  
  
FairGeoMedium* mTof      = geoMedia->getMedium("plasticForTOF");  
if ( ! mTof ) Fatal("Main", "FairMedium plasticForTOF not found");  
geoBuild->createMedium(mTof);  
TGeoMedium* pMed34 = gGeoMan->GetMedium("plasticForTOF");  
if ( ! pMed34 ) Fatal("Main", "Medium plasticForTOF not found");  
// -----
```



Create top volume

```
// ----- Create geometry and top volume -----  
gGeoMan = (TGeoManager*)gROOT->FindObject("FAIRGeom");  
gGeoMan->SetName("TOFgeom");  
TGeoVolume* top = new TGeoVolumeAssembly("TOP");  
gGeoMan->SetTopVolume(top);  
// -----
```



- Create keeping volume

```
TGeoVolumeAssembly *ptof = new TGeoVolumeAssembly("TOF");
```




Shape and volume

- Create shape and then volume by assigning material to it

```
// Shape: T0FBox type: TGeoBBox
dx = 94.450000;
dy = 73.450000;
dz = 0.500000;
TGeoShape *pT0FBox = new TGeoBBox("T0FBox", dx,dy,dz);
// Volume: T0FLog
TGeoVolume *pT0FLog = new TGeoVolume("T0FLog",pT0FBox, pMed34);
```



Add to keeping volume

- Repeat multiple times with different position / rotation and copy number (in case of complex geometry)

```
TGeoCombiTrans *t0 = new TGeoCombiTrans("t0");  
ptof->AddNode(pTOFLog, 0, t0);
```



Add keeping volume to top

- Here the global position and rotation are defined

```
dx = -417.359574;  
dy = 2.400000;  
dz = 960.777114;
```

```
TGeoRotation *gRot = new TGeoRotation();  
gRot->RotateX(0.);  
gRot->RotateY(-31.000000);  
gRot->RotateZ(0.);
```

```
TGeoCombiTrans* pMatrix2 = new TGeoCombiTrans("", dx,dy,dz,gRot);
```

```
top->AddNode(ptof, 0, pMatrix2);
```



- Test the geometry and write it to output file

```
// ----- Finish -----  
gGeoMan->CloseGeometry();  
gGeoMan->CheckOverlaps(0.001);  
gGeoMan->PrintOverlaps();  
gGeoMan->Test();  
  
TFile* geoFile = new TFile(geoFileName, "RECREATE");  
top->Write();  
geoFile->Close();  
// -----
```



Sensitive (active) volumes

- In the corresponding detector class R3BTof.cxx
- Implement virtual function, which checks the name of a volume

```
Bool_t R3BTof::CheckIfSensitive(std::string name)
{
    if (TString(name).Contains("TOFLog"))
    {
        return kTRUE;
    }
    return kFALSE;
}
```

Name assigned in the geometry macro



Accessing geometry (simulation)



Run the standard simulation

- `cd BUILD_DIR`
- `./config.sh`
- `cd r3broot/macros/r3b`
- `root -l r3bsim.C`
- `[..] .q`



Implementation in macro

```
{
  // Open parameter file
  TFile *file = new TFile("r3bpar.root");
  // Get the complete geometry created during simulation
  FairGeoParSet *par = (FairGeoParSet*) file-
>Get("FairGeoParSet");

  // Get the volume with specific name
  TGeoVolume *toflog = gGeoManager-
>FindVolumeFast("TOFLog");
  // Get its shape
  TGeoBBox *box = (TGeoBBox*)toflog->GetShape();
  cout << endl << " half-width x = " << box->GetDX() << "
cm" << endl;
}
```




Implementation in the code (analysis task)

```
#include "FairRuntimeDb.h"
```

```
#include "FairGeoParSet.h"
```

```
#include "TGeoVolume.h"
```

- In the **SetParContainers()** member-function

```
FairRuntimeDb *rtdb = FairRuntimeDb::Instance();
```

```
fgeopar = (FairGeoParSet*) rtdb->getContainer("FairGeoParSet");
```

- In the **Init()** member-function

```
TGeoManager *geo = fgeopar->GetGeometry();
```

```
TGeoVolume *toflog = geo->FindVolumeFast("TOFLog");
```

```
...
```



- Fix in your FairSoft installation (missing geant3 data files)

- ➔ `cd FAIRSOFT_BUILD/share`

- ➔ `mkdir geant3`

- ➔ `cd geant3`

- ➔ `cp -r FAIRSOFT_SOURCE/transport/geant3/data .`

- ➔ `cd R3BROOT_BUILD`

- ➔ `./config.sh`

- ➔ `cmake ../r3broot`