

Adaptation of StiRnD/Hft/  
StiHftDetectorBuilder to use sensor  
placement

# status

- Right now, this maker (used to place the detector volume for the tracking) is using the ROOT geometry
- it uses the path defined the ROOT Geometry

```
TString nameP(nodeP->GetName());

TString temp=nameP;
temp.ReplaceAll("HALL_1/CAVE_1/IDSM_1/PXMO_1/", "");
int q=temp.Index("_");
temp.Replace(0,q+1, "");
TString numsec=temp(0,2);
int sector=numsec.Atoi();
```

- **This will not work for real data**

# Example of the SSD

- StSsdDbMaker writes in another table the rotation and translation matrices of all wafer

```
#include "tables/St_ssdWafersPosition_Table.h"  
St_ssdWafersPosition *ssdwafer = new St_ssdWafersPosition("ssdWafersPosition",NoWafers);
```

- This table is used later when the SSD object is initialize :
  - Loop over ladder
  - Loop over wafer
  - Building of the StSsdWafer object which contains these matrices
- I'm not sure if such classes/structures exist for the PXL so the simplest is to implement a similar table (St\_pxlSensorPosition) :
  - Which will be filled in StPxlDbMaker
  - Which values will be accessible directly