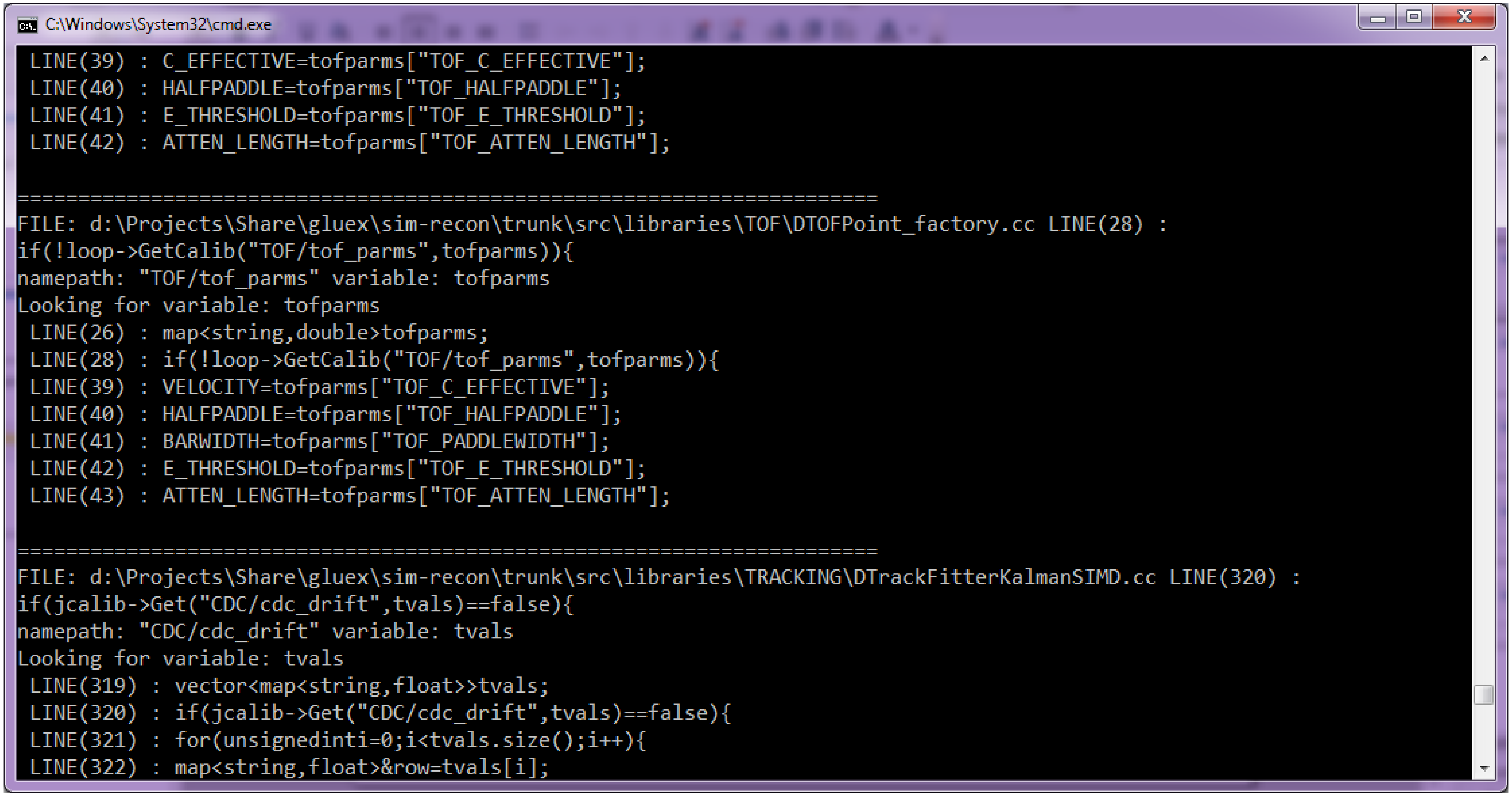
CCDB code verifier



C:\Windows\System32\cmd.exe

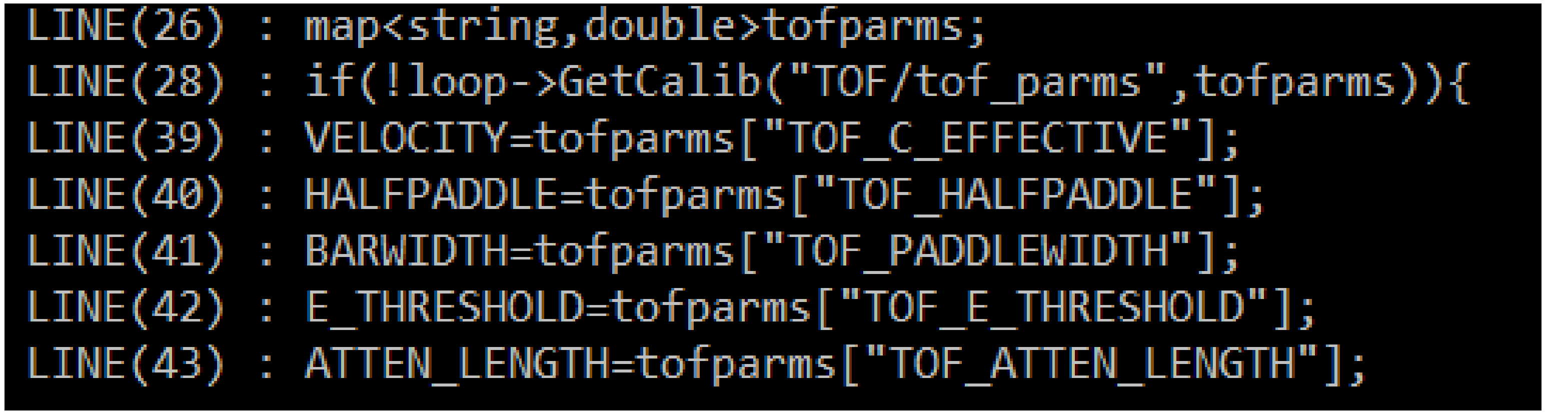
```
 LINE(39) : C_EFFECTIVE=tofparms["TOF_C_EFFECTIVE"];
 LINE(40) : HALFPADDLE=tofparms["TOF_HALFPADDLE"];
 LINE(41) : E_THRESHOLD=tofparms["TOF_E_THRESHOLD"];
 LINE(42) : ATTEN_LENGTH=tofparms["TOF_ATTEN_LENGTH"];


========================================================================
FILE: d:\Projects\Share\gluex\sim-recon\trunk\src\libraries\TOF\DTOFPoint_factory.cc LINE(28) :
if(!loop->GetCalib("TOF/tof_parms",tofparms)){
namepath: "TOF/tof_parms" variable: tofparms
Looking for variable: tofparms
 LINE(26) : map<string,double>tofparms;
 LINE(28) : if(!loop->GetCalib("TOF/tof_parms",tofparms)){
 LINE(39) : VELOCITY=tofparms["TOF_C_EFFECTIVE"];
 LINE(40) : HALFPADDLE=tofparms["TOF_HALFPADDLE"];
 LINE(41) : BARWIDTH=tofparms["TOF_PADDLEWIDTH"];
 LINE(42) : E_THRESHOLD=tofparms["TOF_E_THRESHOLD"];
 LINE(43) : ATTEN_LENGTH=tofparms["TOF_ATTEN_LENGTH"];


========================================================================
FILE: d:\Projects\Share\gluex\sim-recon\trunk\src\libraries\TRACKING\DTrackFitterKalmanSIMD.cc LINE(320) :
if(jcalib->Get("CDC/cdc_drift",tvals)==false){
namepath: "CDC/cdc_drift" variable: tvals
Looking for variable: tvals
 LINE(319) : vector<map<string,float>>tvals;
 LINE(320) : if(jcalib->Get("CDC/cdc_drift",tvals)==false){
 LINE(321) : for(unsignedinti=0;i<tvals.size();i++){
 LINE(322) : map<string,float>&row=tvals[i];
```

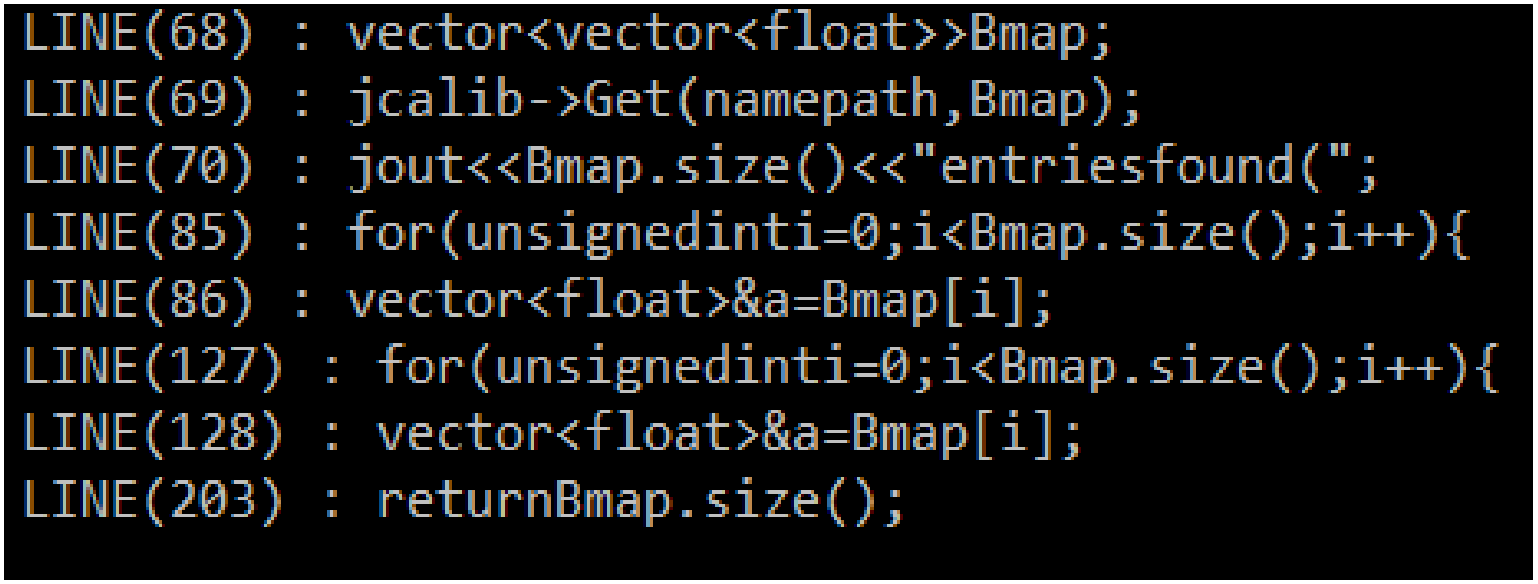
Two ways of use Get

- vector<map<...,...> >



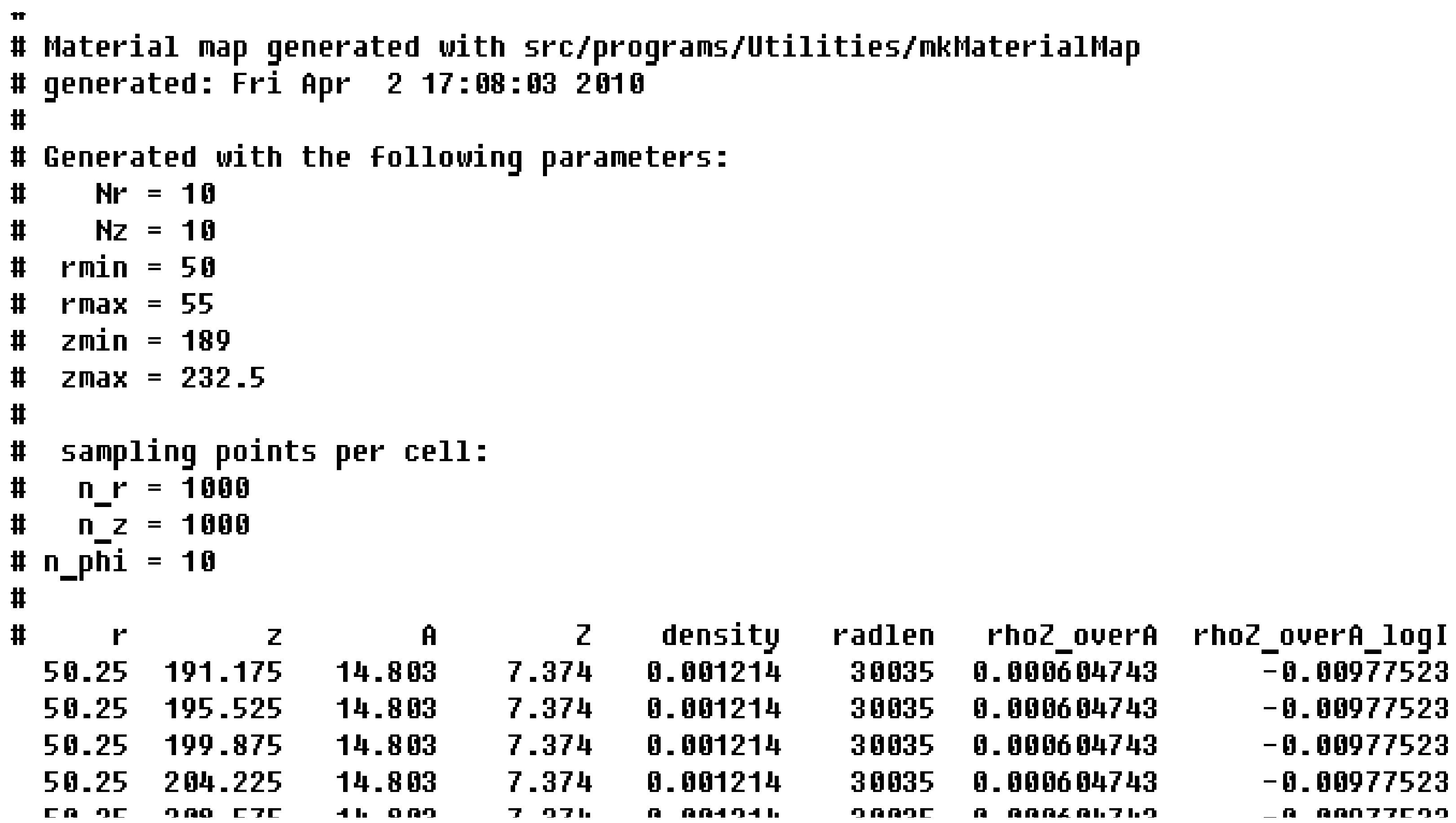
```
LINE(26) : map<string,double>tofparms;
LINE(28) : if(!loop->GetCalib("TOF/tof_parms",tofparms)){
LINE(39) : VELOCITY=tofparms["TOF_C_EFFECTIVE"];
LINE(40) : HALFPADDLE=tofparms["TOF_HALFPADDLE"];
LINE(41) : BARWIDTH=tofparms["TOF_PADDLEWIDTH"];
LINE(42) : E_THRESHOLD=tofparms["TOF_E_THRESHOLD"];
LINE(43) : ATTEN_LENGTH=tofparms["TOF_ATTEN_LENGTH"];
```

- vector<vector<...>>



```
LINE(68) : vector<vector<float>>Bmap;
LINE(69) : jcalib->Get(namepath,Bmap);
LINE(70) : jout<<Bmap.size()<<"entriesfound(";
LINE(85) : for(unsignedinti=0;i<Bmap.size();i++){
LINE(86) : vector<float>&a=Bmap[i];
LINE(127) : for(unsignedinti=0;i<Bmap.size();i++){
LINE(128) : vector<float>&a=Bmap[i];
LINE(203) : returnBmap.size();
```

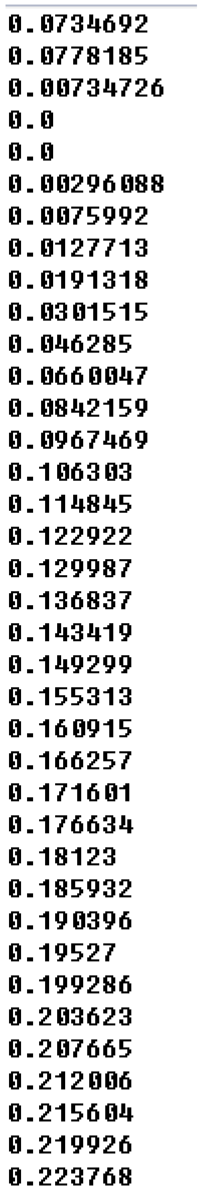
Column names mess



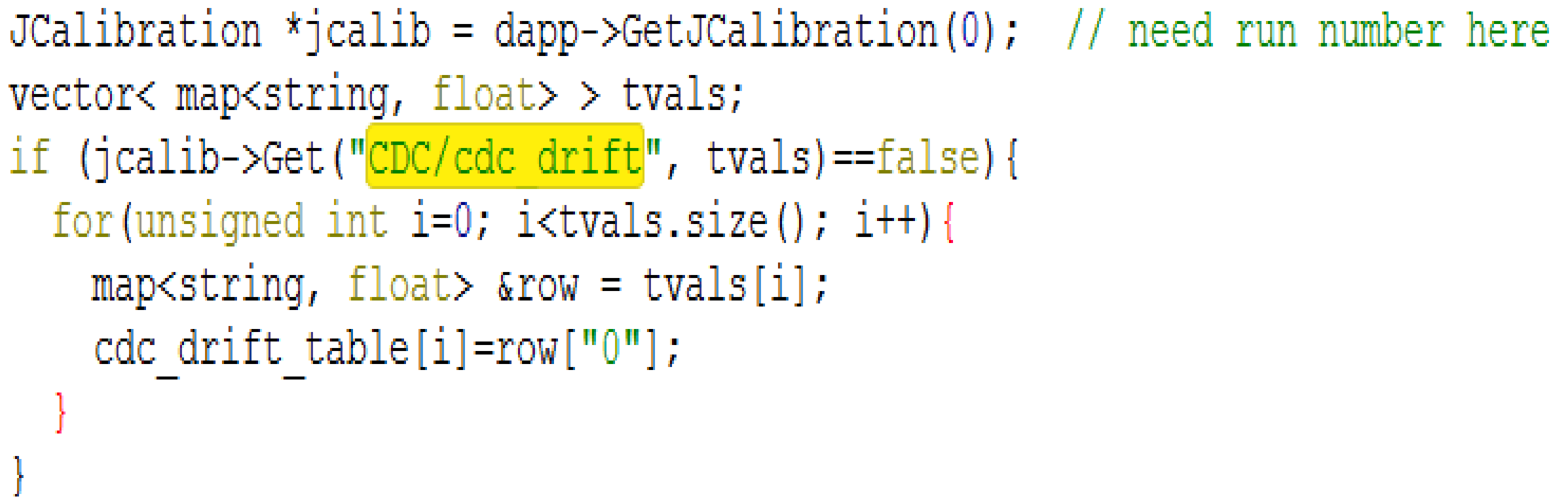
```
#
# Material map generated with src/programs/Utilities/mkMaterialMap
# generated: Fri Apr  2 17:08:03 2010
#
# Generated with the following parameters:
#    Nr = 10
#    Nz = 10
#  rmin = 50
#  rmax = 55
#  zmin = 189
#  zmax = 232.5
#
#  sampling points per cell:
#   n_r = 1000
#   n_z = 1000
# n_phi = 10
#
#     r        z         A         Z    density    radlen   rhoZ_overA  rhoZ_overA_logI
  50.25  191.175    14.803     7.374   0.001214     30035  0.000604743      -0.00977523
  50.25  195.525    14.803     7.374   0.001214     30035  0.000604743      -0.00977523
  50.25  199.875    14.803     7.374   0.001214     30035  0.000604743      -0.00977523
  50.25  204.225    14.803     7.374   0.001214     30035  0.000604743      -0.00977523
  50.25  208.575    14.803     7.374   0.001214     30035  0.000604743      -0.00977523
```

- JANA requires #% to identify string as column names string

Example of usage



```
0.0734692
0.0778185
0.00734726
0.0
0.0
0.00296088
0.0075992
0.0127713
0.0191318
0.0301515
0.046285
0.0660047
0.0842159
0.0967469
0.106303
0.114845
0.122922
0.129987
0.136837
0.143419
0.149299
0.155313
0.160915
0.166257
0.171601
0.176634
0.18123
0.185932
0.190396
0.19527
0.199286
0.203623
0.207665
0.212006
0.215604
0.219926
0.223768
```



```
JCalibration *jcalib = dapp->GetJCalibration(0);  // need run number here
vector< map<string, float> > tvals;
if (jcalib->Get("CDC/cdc_drift", tvals)==false){
  for(unsigned int i=0; i<tvals.size(); i++){
    map<string, float> &row = tvals[i];
    cdc_drift_table[i]=row["0"];
  }
}
```

Conclusion

- Please,
- Use vector<map<...,...> > if column names are used
- Use vector<vector<...>> if you want to have just tabled data
- Don't forget #% sign while using JANA SVN calibration
- Thank you!