Database complementary information :  
  
sbs\_ft.tracker.tmp\_comm\_range =200

sbs\_ft.tracker.commonmode\_groupsize = 128

Output file :

Spectrometer name:

sbs

Detector name:

gems  
  
variables:

Plane variables:

(reminder: in this context, a ‘‘plane’’ is a readout directionper chamber)

e.g.

*sbs.gems.x1.\** variables for x1 plane

Plane variables also contains coordinate objects, which stores the information of the track wrt the plane

Coordinate variables:

*sbs.gems.x1.coord.\**

*pos*: position of the 2D track on the given plane

*resid*: residual of the cluster position wrt the position of the 2D track on the given plane

*slope*: slope of the 2D track on the given plane

*3Dpos*: position of the 3D track on the given plane

*3Dresid*: residual of the cluster position wrt the position of the 3D track on the given plane

*3Dslope*: slope of the 3D track on the given plane

Projection variables (projection x is the combination of all x planes)

*sbs.gems.x.\**  
  
In the projection are fitted the 2D tracks or ‘‘roads’’

Road variables:

*sbs.gems.x.rd.\**

*chi2, ndof*: varaibles for road fitting  
*good*: flag to indicate if this track is considered good for the algorithm

*pos*: position of the 2D track at z = 0

*slope*: slope of the 2D track

*trkstat*: tracking status as defined in ‘‘Road.h’’

enum ETrackingStatus {

kTrackOK = 0,

kTooFewPlanesWithHits = 1, // Too few planes with hits

kTooManyHitCombos = 2, // Too many hit combinations to fit

kNoGoodFit = 3, // No fit with good chi2

};