class MvdDetector : public CbmDetector

MvdDetector();
MvdDetector(const char* name, Bool_t active);
virtual ~MvdDetector();

virtual void Initialize();
virtual Bool_t ProcessHits(CbmVolume* vol = 0);
virtual void EndOfEvent();
virtual void   FinishRun();
virtual void Register();

virtual TClonesArray* GetCollection(Int_t iColl) const;
virtual void Print() const;
virtual void Reset();
virtual void CopyClones(TClonesArray* cl1, TClonesArray* cl2,
Int_t offset);
virtual void ConstructGeometry();
void ConstructRootGeometry();
void ConstructASCIIGeometry();
void ExpandNode(TGeoNode *fN);

class MvdPoint : public CbmMCPoint

MvdPoint();
MvdPoint(Int_t trackID, Int_t detID, TString detName,
TVector3 posIn, TVector3 posOut,
TVector3 momIn, TVector3 momOut,
Double_t tof, Double_t length, Double_t eLoss);
MvdPoint(const MvdPoint& point)
virtual ~MvdPoint();

Double_t GetXOut() const
Double_t GetYOut() const
Double_t GetZOut() const
Double_t GetPxOut() const
Double_t GetPyOut() const
Double_t GetPzOut() const
TString GetDetName() const
void PositionOut(TVector3& pos)
void MomentumOut(TVector3& mom)

void SetPositionOut(TVector3 pos);
void SetMomentumOut(TVector3 mom);

virtual void Print(const Option_t* opt) const;

class MvdDigiTask : public CbmTask

MvdDigiTask();

MvdDigiTask(Double_t lx, Double_t ly, Double_t treshold,
Double_t noise, Double_t slx, Double_t sly, Double_t streshold,
Double_t snoise);

virtual ~MvdDigiTask();

virtual void SetParContainers();
virtual InitStatus Init();
virtual InitStatus ReInit();

virtual void Exec(Option_t* opt);

class MvdIdealRecoTask : public CbmTask

MvdIdealRecoTask();
MvdIdealRecoTask(Double_t sx, Double_t sy, Double_t sz);

~MvdIdealRecoTask();

virtual void SetParContainers();
virtual InitStatus Init();

virtual void Exec(Option_t* opt);

Strip/Pixel

class MvdHybridHitProducer : public CbmTask

MvdHybridHitProducer();
MvdHybridHitProducer(Double_t lx, Double_t ly, Double_t
treshold, Double_t noise);
virtual ~MvdHybridHitProducer();

virtual void SetParContainers();
virtual InitStatus Init();
virtual InitStatus ReInit();

virtual void Exec(Option_t* opt);

class MvdStripHitProducer : public CbmTask

MvdStripHitProducer();
MvdStripHitProducer(Double_t topPitch, Double_t botPitch,
Double_t ori, Double_t skew, TVector2 topAnchor, TVector2
botAnchor, Int_t nrTopFE, Int_t nrBotFE, Int_t nrFECh, Double_t
threshold, Double_t noise);
virtual ~MvdStripHitProducer();

virtual void SetParContainers();

virtual InitStatus Init();
virtual InitStatus ReInit();

virtual void Exec(Option_t* opt);

class MvdDigiPixel : public Tobject

MvdDigiPixel();
MvdDigiPixel(Int_t index, Int_t detID, TString detName,
Int_t fe, Int_t col, Int_t row, Double_t charge);
~MvdDigiPixel()

Int_t GetFE() const
Int_t GetPixelColumn() const
Int_t GetPixelRow() const
TString GetDetName() const
Double_t GetCharge() const
Int_t GetDetID() const
Int_t GetIndex() const

class MvdDigiStrip : public Tobject

MvdDigiStrip();
MvdDigiStrip(Int_t index, Int_t detID, TString detName,
Int_t fe, Int_t chan, Double_t charge);
~MvdDigiStrip()

Int_t GetIndex() const
Int_t GetDetID() const
TString GetDetName() const
Int_t GetFE() const
Int_t GetChannel() const
Double_t GetCharge() const
Int_t GetMCID() const

void SetIndex (Int_t index)
void SetDetID (Int_t detID)
void SetDetName (const TString& detName)
void SetFE (Int_t fe)
void SetChannel (Int_t channel)
void SetCharge (Double_t charge)
void SetMCID (Int_t mcID)

void print();

class MvdSimplePixelClusterFinder : public MvdClusterFinder
MvdSimplePixelClusterFinder():MvdClusterFinder()
MvdSimplePixelClusterFinder(std::vector<double> params,
std::vector<MvdDigiPixel> hits)
std::vector< std::vector< MvdDigiPixel> > GetClusters();
MvdDigiPixel MoveHit(std::vector<MvdDigiPixel>* hitVector, int index);
bool IsInRange(MvdDigiPixel hit1, MvdDigiPixel hit2);

class MvdClusterCand : public TObject

MvdClusterCand()
~MvdClusterCand()
MvdClusterCand(std::vector<int> list);
void SetClusterList(std::vector<int> list)

std::vector<int> GetClusterList()
int GetClusterSize()

void Print();

class MvdChargeWeightedPixelMapping : public MvdPixelBackMapping

MvdChargeWeightedPixelMapping(std::vector<MvdDigiPixel> pixelArray,
std::vector<double> params):MvdPixelBackMapping(pixelArray, params)
MvdCluster GetCluster();
TGeoHMatrix GetTransformation(std::string detName);
TVector3 GetSensorDimensions(std::string detName);

class MvdHit : public CbmHit

MvdHit();
MvdHit(Int_t trackID, Int_t detID, TString detName,
TVector3& pos, TVector3& dpos, Int_t index, Double_t
charge, Int_t NPixelHits);
virtual ~MvdHit();

void SetTrackID(Int_t id)
void SetDetName(TString name)
void SetCharge(Double_t charge)
void SetNPixelHits(Int_t pixel)

TString GetDetName()const
Int_t GetTrackID()
Double_t GetCharge()
Int_t GetNPixelHits() const
TVector3 GetPosition() const

virtual void Print(const Option_t* opt = 0) const;
TString fDetName; // Detector name
Double_t fCharge;
Int_t fNPixelHits;
ClassDef(MvdHit,6);