Track and Related MDST classes

Track

Public Member Functions

	Track () Constuctor without arguments; needed for I/O.
	~Track () Destructor.
const TrackFitResult *	**getTrackFitResult** (const **Const::ChargedStable** &chargedStable) const Access to TrackFitResults.
void	**setTrackFitResultIndex** (const **Const::ChargedStable** &chargedStable, short index) Set an index (for positive values) or unavailability-code (with negative values) for a specific mass hypothesis.

Charged Stable maybe replaced with other class, but currently (pi, K, p, e, mu)

Private Attributes

All indices can be the same, e.g. for high momentum track.

short int	**m_trackFitIndices** [5] Index list of the TrackFitResults associated with this **Track**.

TrackFitResult

Public

	TrackFitResult () Constructor for I/O purpuses.
TVector3	**getPosition** () const Constructor taking an object used during the fitting process.
void	**setPosition** (const TVector3 &position)
TVector3	**getMomentum** () const
void	**setMomentum** (const TVector3 &momentum)
TMatrixF	**getCovariance6** () const Position and Momentum Covariance Matrix.
void	**setCovariance6** (const TMatrixF &covariance)
onst::ParticleType	**getParticleType** () const Get back a ParticleCode of the hypothesis of the track fit.
void	**setParticleType** (const Const::ParticleType &pType)
short	**getCharge** () const Return track charge (1 or -1).
void	**setCharge** (int charge) Setter for Charge.
float	**getPValue** () const Getter for Chi2 Probability of the track fit.
void	**setPValue** (float pValue)

TrackFitResult private

Private Attributes

unsigned int	**m_pdg** Number of **PXD** hits used in the **TrackFitResult**.
float	**m_pValue** Chi2 Probability of the fit.
short	**m_charge**

+ helix at the perigee.
We just make a magnetic field up. This works probably for any real magnetic field, but with nonsensical values for pT.
Need to save B Field for getting correct helix; different interpretation for 0-field.

HitPatternCDC

https://belle2.cc.kek.jp/browse/viewvc.cgi/svn/trunk/software/tracking/dataobjects/include/HitPatternCDC.h?revision=8896&view=markup

- 64 bit bitset as actual saver of hit pattern
  - each layer
  - for Super-Layers 2-9 (1-8 if you start counting at 0)

    there is information, if there was at least one layer with 2 hits.

HitPatternVXD

- Saving 0-2 hits of the **outgoing arm**.
- Definitively for masked mode separately.
- I’m not yet sure, if we need separate count for rescue mechanism as well.

- Everything fits into 16 bits