Tracking -- Propagation

- Goal: Propagate and store track parameters & covariance matrices at each sensor
  - Calculate and store during GBL
  - Store as persistent TrackStates, which are added to existing List<TrackState> associated with GBLTrack
  - Maintain full backwards-compatibility
- Collapse into a single issue:
  - **154 Track parameter and covariance propagation**
  - 54 Propagate covariance matrix correctly when changing reference point for perigee track representation
  - 158 MS covariance matrix in HpsGblRefitter
- As cross-check of results: driver that (re-)calculates track params & covariances using existing GBL output
  - Already started by Bradley in

    11 A track should have track states at every SVT layer and the Ecal

Rationale

- Why calculate & store during GBL?
  - Track params already calculated at each sensor for GBL fits
  - For covariances, code stumps exist that can be completed
  - Re-calculating after GBL is computationally expensive!
- Why at each sensor, rather than each layer?
  - This is where GBL does its fits anyway
  - Allows straightforward plotting of residuals
  - If this uses too much memory/disk space, can later decide to only store states at some sensors / for some types of tracks / for calibrations
- What about track state at a vertex?
  - Since one track can be used in multiple vertices, vertex-refitted params & covariances should be associated with the Vertex not the Track
  - Will be addressed in another part of the code
  - Tackle as a separate Issue at a later time

Using TrackState Object

- Why use TrackStates?
  - TrackState object has everything we need
  - List<TrackState> associated with a Track is of arbitrary length, so is easy to lengthen without breaking backwards-compatibility
- Each TrackState has a location code
  - Currently store TrackStates only AtIP, AtLastHit, AtCalorimeter
  - Assign new TrackStates to all have location code 0, write new method to fetch TrackState corresponding to a given sensor#

```
// TrackState location codes.
public final static int AtOther = 0;
public final static int AtIP = 1;
public final static int AtFirstHit = 2;
public final static int AtLastHit = 3;
public final static int AtCalorimeter = 4;
public final static int AtVertex = 5;
public final static int LastLocation = AtVertex;
```