TkrRecon
Iterative Recon

• What is it?
  - The Iterative recon is a mechanism for allowing parts of the Tracker Reconstruction software to be called more than once per event. In particular, existing pattern recognition tracks can be refit and the vertex algorithm rerun.

• Why is it needed?
  - The Calorimeter needs/wants the output of TkrRecon in order to refine the initial energy determination.
  - The TkrRecon wants the best energy estimate in order to get the best track fits and, subsequently, the best vertices.
  - The Iterative Recon solves this problem by providing Cal Recon with tracking information to get an improved energy for an event which is then used to refit the tracks and rerun the vertexing.
  - This process can be repeated as many times as the user likes.
TkRecon
Iterative Recon
Current Reconstruction

CalRecon
First Pass

TkRecon

CalRecon
Second Pass

CalXtralRecAlg
CalClustersAlg
TkReconAlg
TkClstersAlg
TkFindAlg
TkTrackFitAlg
TkVertexAlg
CalClustersAlg
Iterative Reconstruction

- Two Assumptions:
  - The results of the Second Pass CalRecon (ie the improved energy determination) are obtained from the CalClusters via the method “getEnergyCorrected”
  - Tkr Clustering and Pattern Recognition do not need to be rerun.

- Basic idea:
  - Find the CalClusters and current Pattern/Fit Tracks in the TDS
  - Reassign Calorimeter energy to the current set of Fit Tracks
    - Algorithm is currently identical to that used in the Combo Pat Rec
    - Only assigns energy to the two best tracks...
  - Rerun the vertexing
TkrRecon
Iterative Recon

CalRecon First Pass

TkrRecon First Pass

CalRecon First Pass

CalRecon First Pass

TkrReconAlg

TkrTrackFitAlg

TkrVertexAlg
TkrRecon
Iterative Recon

• How to make it work
  - Modify the Reconstruction part of the basic_options file in Gleam:

```plaintext
Reconstruction.Members={
  "Sequencer/Cal1",
  "Sequencer/Tkr",
  "Sequencer/Cal2",
  "Sequencer/TkrIter",
  "Sequencer/Acd"
};

Cal1.Members = {
  "CalXtalRecAlg",
  "CalClustersAlg/first"
};
Tkr.Members = {
  "TkrReconAlg/FirstPass"
};
Cal2.Members = {
  "CalClustersAlg/second"
};
TkrIter.Members = {
  "TkrReconAlg/Iteration"
};
```