

# **TkrRecon Reorganization Update**

**Analysis Group Meeting**

**Nov 22, 2004**

Introduction

Overview of the new TkrRecon TDS Classes

Overview of changes in the TkrRecon and TkrUtil Packages

Status

# TkrRecon Reorganization Introduction

- TkrRecon circa GlastRelease v4r6
  - It Works!
- But...it all consists of
  - A top level of Gaudi Algorithms, Subalgorithms and Tools
  - All worked into a form to utilize an array of non-Gaudi classes to do the work
    - Which have mostly evolved from the original pre-Gaudi classes
  - With too much near duplication of code
  - All tied to an complicated TDS structure
    - Which doesn't work properly with the hoped move to "converters"
- Further
  - Not a flexible structure
    - e.g. can't change particle hypothesis without rebuilding nearly all of Glean
  - Even the experts have difficulty maintaining it

# TkrRecon Reorganization

## Introduction: Goals

- Simplify the overall TkrRecon package
  - Reorganize the code
    - Move the “work” into Gaudi tools – reduce overall number of classes
    - Consolidate nearly identical classes, more re-use of a classes
    - Move “utility” classes into TkrUtil for general Glem use
  - Ease maintenance for current developers
  - Reduce buy in cost to future new developers
- Simplify output of TkrRecon (TDS & PDS)
  - Reduce number of output classes
    - TkrCluster, TkrTrack/TkrTrackHit/TkrTrackParams, TkrVertex
    - Now stored in Gaudi ObjectContainers
  - Simplify
    - Conventions (e.g. replace Tower, Layer, View with idents::TkrlId)
    - etc.
- Include improvements that have long been on wish lists
  - Monte Carlo pattern recognition (for pat rec comparisons)
  - Change fit track particle hypothesis “on the fly” (e.g. muons)
  - Implement track finding in both “up” and “down” directions
  - etc.

# TkrRecon Reorganization

## Introduction: Plan for Implementation

- Step 1
  - Implement new TDS classes
- Step 2
  - Make existing TkrRecon work with new TDS classes
- Step 3
  - Begin reorganization of TkrRecon and TkrUtil packages
    - Change to generic fit
    - Reorganize Combo Pat Rec into Gaudi Tool, using generic fit
    - Implement new “helper” tools, classes in TkrUtil Done
      - TkrQueryClustersTool
      - TkrTrackParams, TkrCovMatrix (to implement math in track parameter TDS)
      - etc.

---

- Step 4 In Progress
  - Iterate through steps 1 and 3 to “get it right”

---

- Step 5 To Do
  - Final steps
    - Vertexing
    - PDS output

# TDS Classes:

## TkrCluster/TkrClusterCol

- Old:
  - TkrCluster
  - TkrClusterCol
    - `std::vector<TkrCluster*> m_clusterList;`
    - `std::vector<TkrCluster*> m_clustersByPlaneList[NVIEWS][NPLANES];`
- New :
  - TkrCluster
    - `typedef ObjectVector<TkrCluster> TkrClusterCol;`
    - `typedef SmartRefVector<TkrCluster> TkrClusterVec;`
  - Helpers (in TkrUtil/TkrQueryClustersTool)
    - `TkrClusterVec clusters = getClusters(TkrId)` or `(view, layer)`
      - Map generated when clusters are made
      - Called by TkrPoint to generate a list of “space points”

# TDS Classes: TkrCluster

## Common

- start, end strip
- position (float, maybe?)

## Old

- identified by internal id
- layer
- view
- “shared” flag
- Raw ToT

## New

- identified by pointer to object
- TkrId
- status bits
  - shared, layer, view
  - ??
- ToT place-holder for now, real ToT info will be elsewhere

# TDS Classes: Overview of Track Objects

- **Old**

- TkrFitHit
- TkrFitMatrix
- TkrFitPar
- TkrFitPlane
- TkrFitTrackBase
- TkrKalFitTrackBase
- TkrKalFitTrack
- TkrPatCand
- TkrPatCandHit
- TkrRecInfo
- TkrTrackTab

- **New**

- TkrTrackParams
- TkrTrackHit
- TkrTrack

# TDS Classes: TkrTrack

- ... is a SmartRefVector of TkrTrackHits
  - with overall track information
  - many new status bits
- ... is generated by patrec, then the same object is fit by track fitter
  - status bits tell at what stage the track is.



## TDS Classes: TkrTrackHit

- Contains several sets of track parameters (measured, predicted, filtered, smoothed, reverse-filtered, multiple-scattering contribution)
- Overall information about the hit
- Access to parameter information by whether in the measured or non-measured direction, as well as whether x or y:

    getMeasuredPosition(),

instead of

    x = ( XMeasured ? getXPosition() : getYPosition() );

# TDS Classes: TkrTrackParams

- Combines the information in the old TkrFitPar and TkrFitMatrix
- Info can be accessed by index or by name

# Changes to TkrUtil: TkrGeometrySvc

- Convert completely to standard LAT labeling
  - numbering from Grid up
- Geometry service learns about tracker using the propagator.
  - planes and layers are determined from the geometry
  - standard tray/bottom tray
  - x tray on bottom
  - ???
- New access methods by TkrId

## TkrQueryClustersTool

- TkrUtil/Tool and TkrRecon/Class Consolidated
- Methods added to return vector of Clusters, etc.

# Overview of Code Changes

- Combo Pattern Recognition Interfaced to new Clusters Classes with re-write of 3D space point generator
- Pattern recognition interfaced to new TDS Classes TkrTrack & TrkTrackHit
- Kalman Follower-Finder completely re-written taking advantage of new TDS classes and new Kalman Fitter Tool
- Kalman Fitting re-written in a generic form
- Energy Loss mechanism separated
- Forward – Backwards tracking made possible
- Information on all planes kept
- Gaps & dead strips can participate in the fit

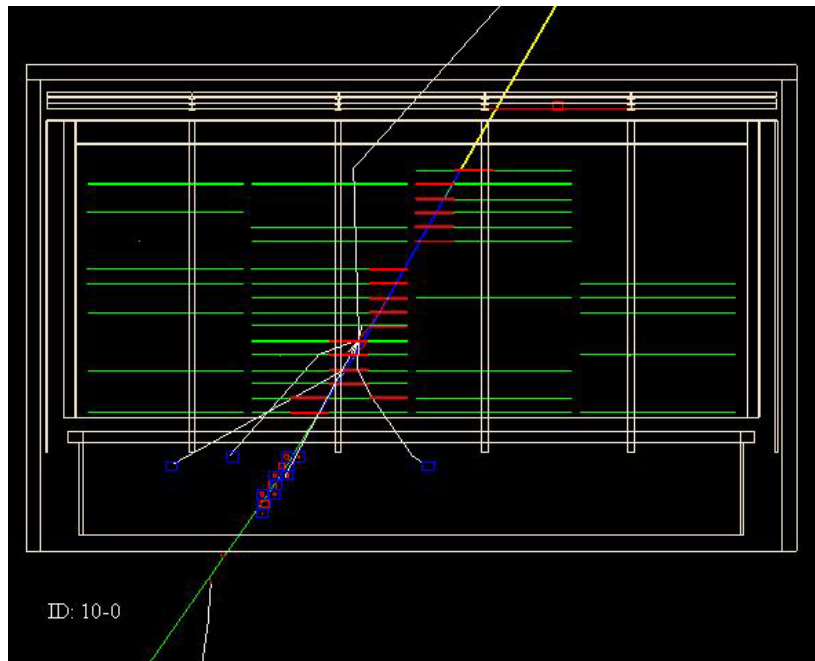
Many thousands of lines of code re-written

Vestigial classes eliminated

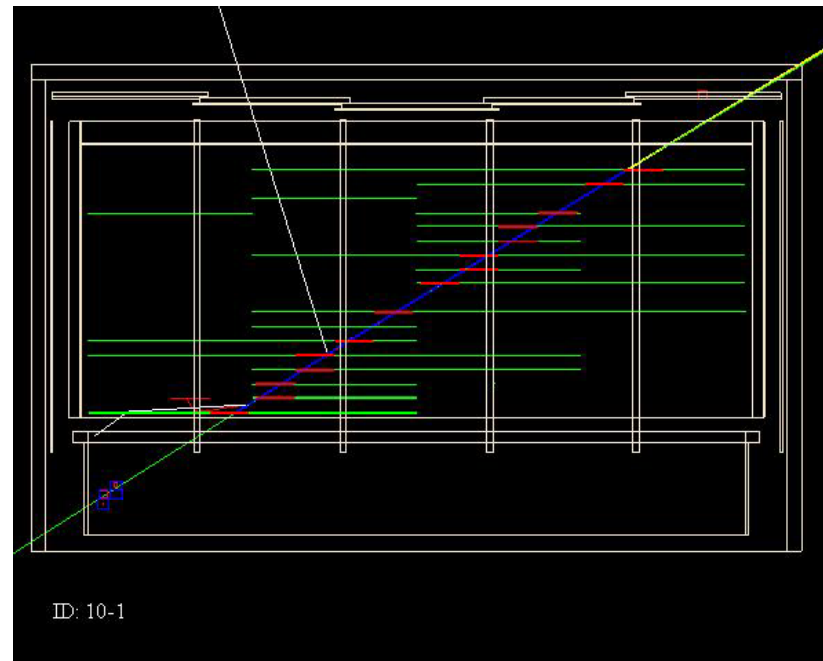
All control parameters user accessible

# 1 GeV $\mu$ - 1 Event Display Pictures

30°  $\mu$  - 1 GeV

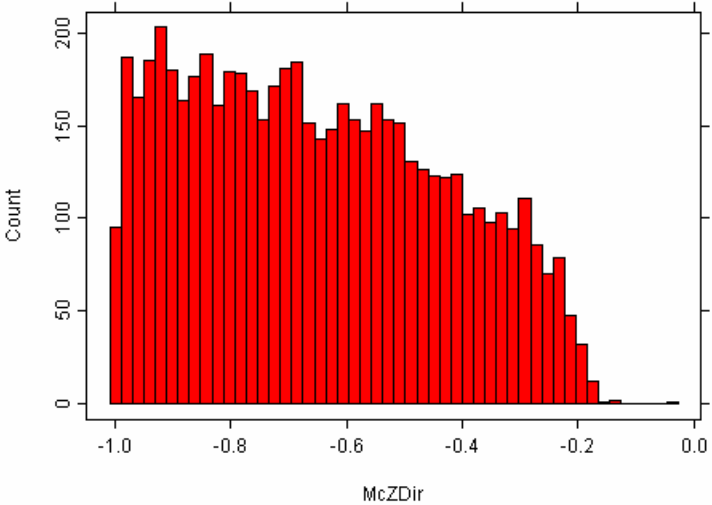


60°  $\mu$  - 1 GeV

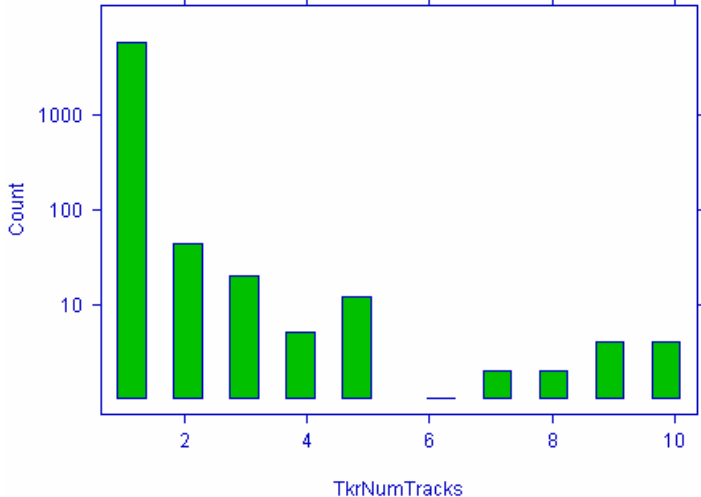


# First 1 GeV $\mu$ Results

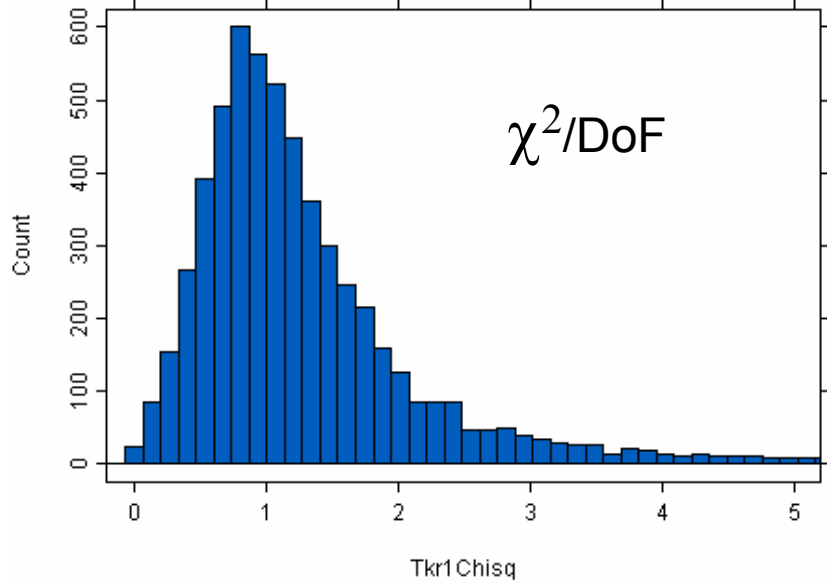
## Acceptance



## Track Count

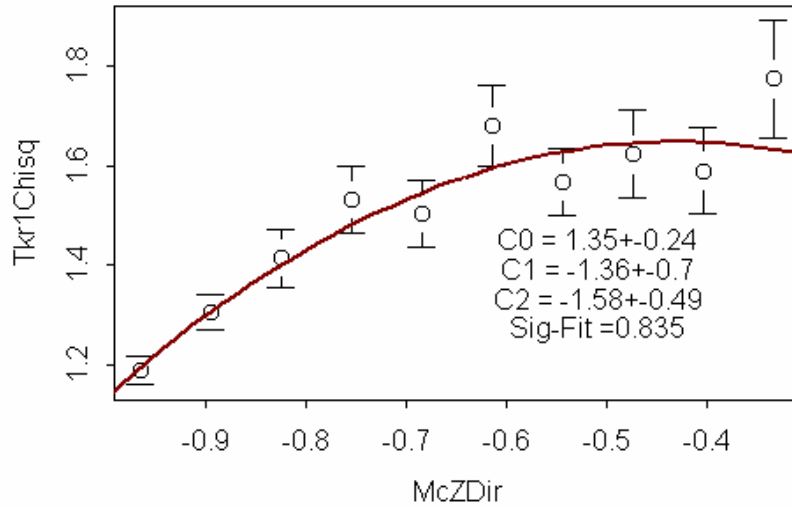


Using new feature  
of energy loss  
mechanism

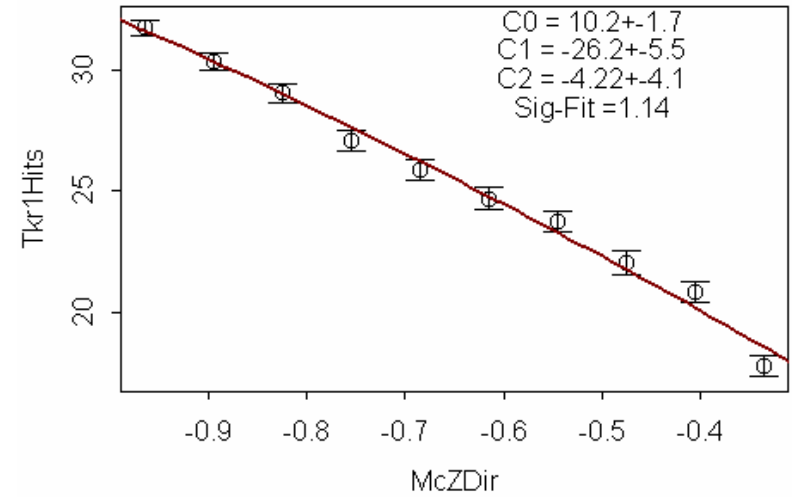


# More 1 GeV $\mu$ Distributions

McZDir Dependencies: 3 Parameters



McZDir Dependencies: 3 Parameters



McZDir Dependencies: 3 Parameters

