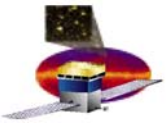


MC Tools Introduction Part I

GLAST Software Analysis Group
Monday, August 12, 2002
Tracy Usher



Root MC Output Classes

(As of August 12, 2002)

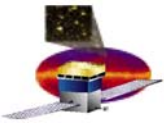
There are four main Monte Carlo Root output classes:

- **McParticle**
 - In "full tree" mode, one for each stable particle produced in a given event
 - Contains basic information on the particle (see later)

- **McPositionHit**
 - One (at least) for each sensitive volume traversed in the tracker
 - Contains basic hit information for these volumes (see later)
 - In the event the particle interacts in a given sensitive volume, separate McPositionHits will be created for each piece of the interaction

- **McIntegratingHit**
 - Calorimeter version of McPositionHit
 - Not used in TkrRecon analysis, no further discussion (sorry!)

- **McEvent**
 - One per event
 - Contains pointers to the lists of the above classes



McParticle

Main Information Access Methods

```
class McParticle : public TObject {
public:
```

```
  Int_t getParticleId() const { return m_particleId;};
```

Particle ID (PDG Value)

```
  /// return an McParticle pointer to the mother particle
```

```
  const McParticle* getMother() const;
```

McParticle pointer to mother

```
  /// return a daughter McParticle corresponding to index
```

```
  const McParticle* getDaughter(Int_t index) const;
```

Access pointer to i^{th} daughter

```
  /// return the full list of daughters
```

```
  const TRefArray& getDaughterList() const { return m_daughters; };
```

List of McParticle pointers to daughters of this particle

```
  Int_t getParticleProperty() const;
```

```
  UInt_t getStatusFlags() const { return m_statusFlags; };
```

```
  /// Retrieve whether this is a primary particle
```

```
  Bool_t primaryParticle() const;
```

```
  /// Retrieve pointer to the start vertex positions
```

```
  const TVector3& getInitialPosition() const;
```

Position of particle at its creation
(global coordinates)

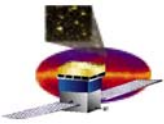
```
  /// Retrieve pointer to end vertex position
```

```
  const TVector3& getFinalPosition() const;
```

Final position of particle - typically
at its stop point
(global coordinates)

```
  const TLorentzVector& getInitialFourMomentum() const;
```

```
  const TLorentzVector& getFinalFourMomentum() const;
```



McPositionHit

Main Information Access Methods

```
class McPositionHit : public TObject {
public:
```

```
const VolumIdentifier& getVolumId() const { return m_volumId; };
```

```
const TVector3& getEntryPosition() const { return m_entry; }
const TVector3& getGlobalEntryPosition() const { return m_entry_global; }
```

```
const TVector3& getExitPosition() const { return m_exit; };
const TVector3& getGlobalExitPosition() const { return m_exit_global; };
```

```
Int_t getMcParticleId() const { return m_mcParticleId; };
```

```
Double_t getDepositedEnergy() const { return m_depositedEnergy; }
```

```
Double_t getParticleEnergy() const { return m_particleEnergy; }
```

```
Double_t getTimeOfFlight() const { return m_timeOfFlight; }
```

```
Double_t getDirectionCosine() const;
```

```
const McParticle* getMcParticle() const {return
(McParticle*)m_mcParticle.GetObject();}
```

```
const McParticle* getOriginMcParticle() const { return
(McParticle*)m_originMcParticle.GetObject();}
```

```
/// Retrieve whether this hit should be digitized
Bool_t needDigi() const;
```

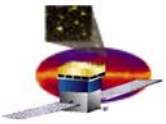
```
};
```

Returns coordinates at entry into the sensitive volume (or at new McParticle start point in case of interaction).
Two methods, one to return LOCAL coordinates, one to return GLOBAL

Returns coordinates at exit point from sensitive volume (or at the point of an interaction).
Two methods, one to return LOCAL coordinates, one to return GLOBAL

Returns McParticle pointer to the particle which caused this McPositionHit.
With this pointer it is possible to trace back up the tree to determine which particular track a hit belongs to.

Returns McParticle pointer to the particle which caused the event to happen.



Augmenting Root MC Classes

▪ Relational tables

- Relate Silicon strip hits/clusters to McPositionHits
- Similarly for Calorimeter
- Coming Soon for Tracker

▪ Private Classes

- McTrack

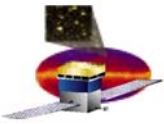
- Contains list of pointers to McPositionHits which contain the same McParticle parent
- List may contain only one pointer (short range delta ray)
- Used to relate list of McPositionHits belonging to individual tracks

- McPair

- Contains pointers to McTracks for THE electron and THE positron resulting from the gamma conversion

- Etc.

- Build upon these two classes to analyze events
- Examples to follow attempt to compare Monte Carlo with TkrRecon
 - Circa Gleam v2r2p4

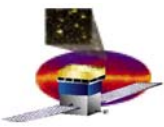


Side Note

- **All Mc root output classes are accessible from the root command line**
 - Need to be sure the mcRootData dll is set up for the root session
 - Run the package setup.bat file before running root
 - Issue a:
`iret = gSystem->Load("%MCROOTDATAROOT%/win32debug/mcRootData.dll");`
from the root command line to load the library

- **Auxiliary classes also accessible in root**
 - Use as part of a larger macro
 - Use as part of another dll accessible to root

- **Also running from Visual Studio**
 - Earlier problems were due to a dll mismatch
 - But prefer to debug in this environment, then run from command line



Example MC Track Plots

MC Gammas

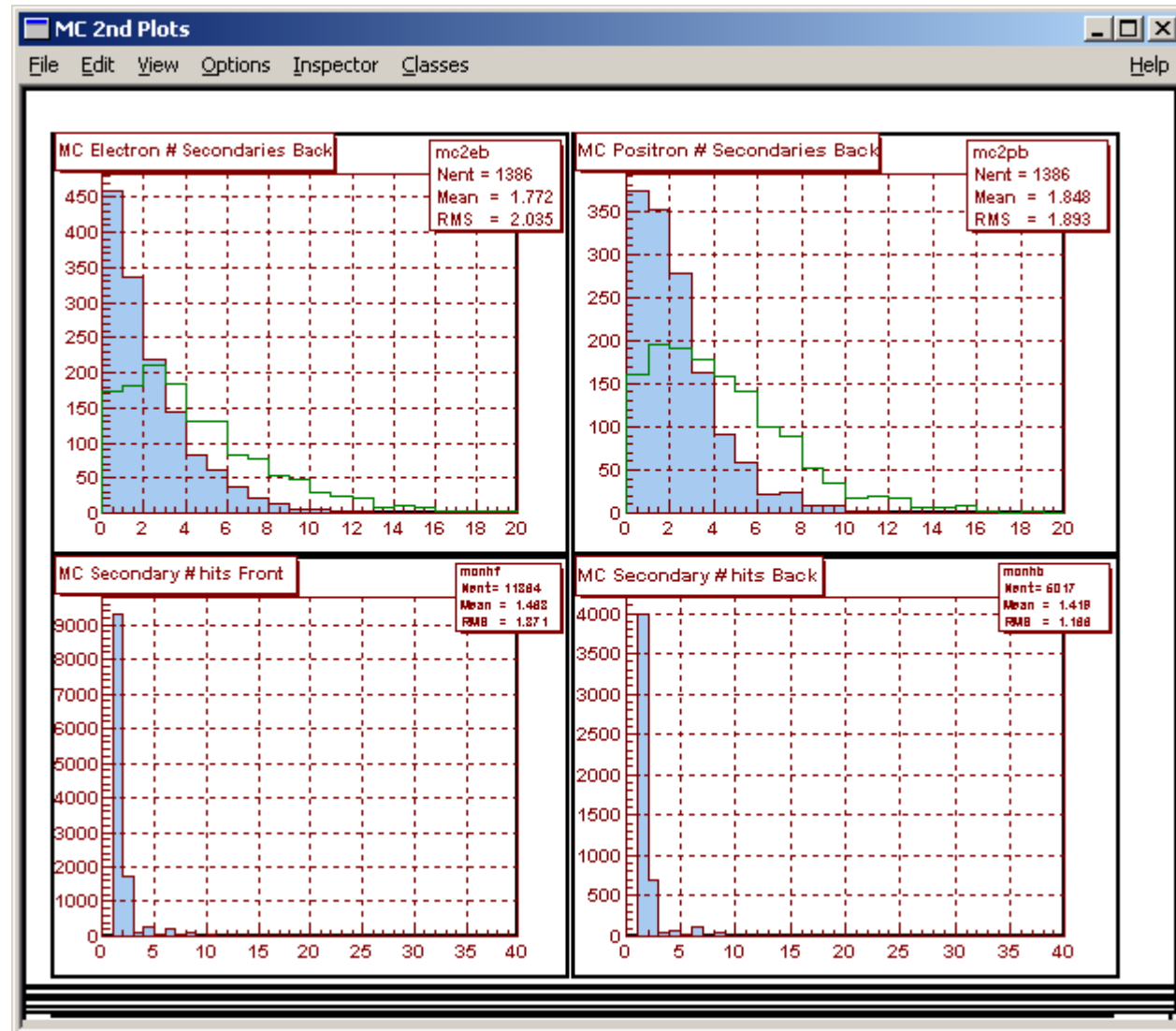
- Energy = 100 MeV
- Downward going

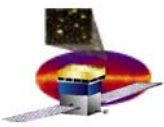
- Top row: Number of secondary “McTracks” attached to primary electron (left column) or positron (right column).

- Blue shaded is Back section
- Green line is Front section

- Bottom row: The number of McPositionHits associated with each McTrack

- Dominated by one and two hits
- Very short range delta rays
- Resolution problem for track hits?





Example MC Track Plots

MC Gammas

- Energy = 100 MeV
- Downward going

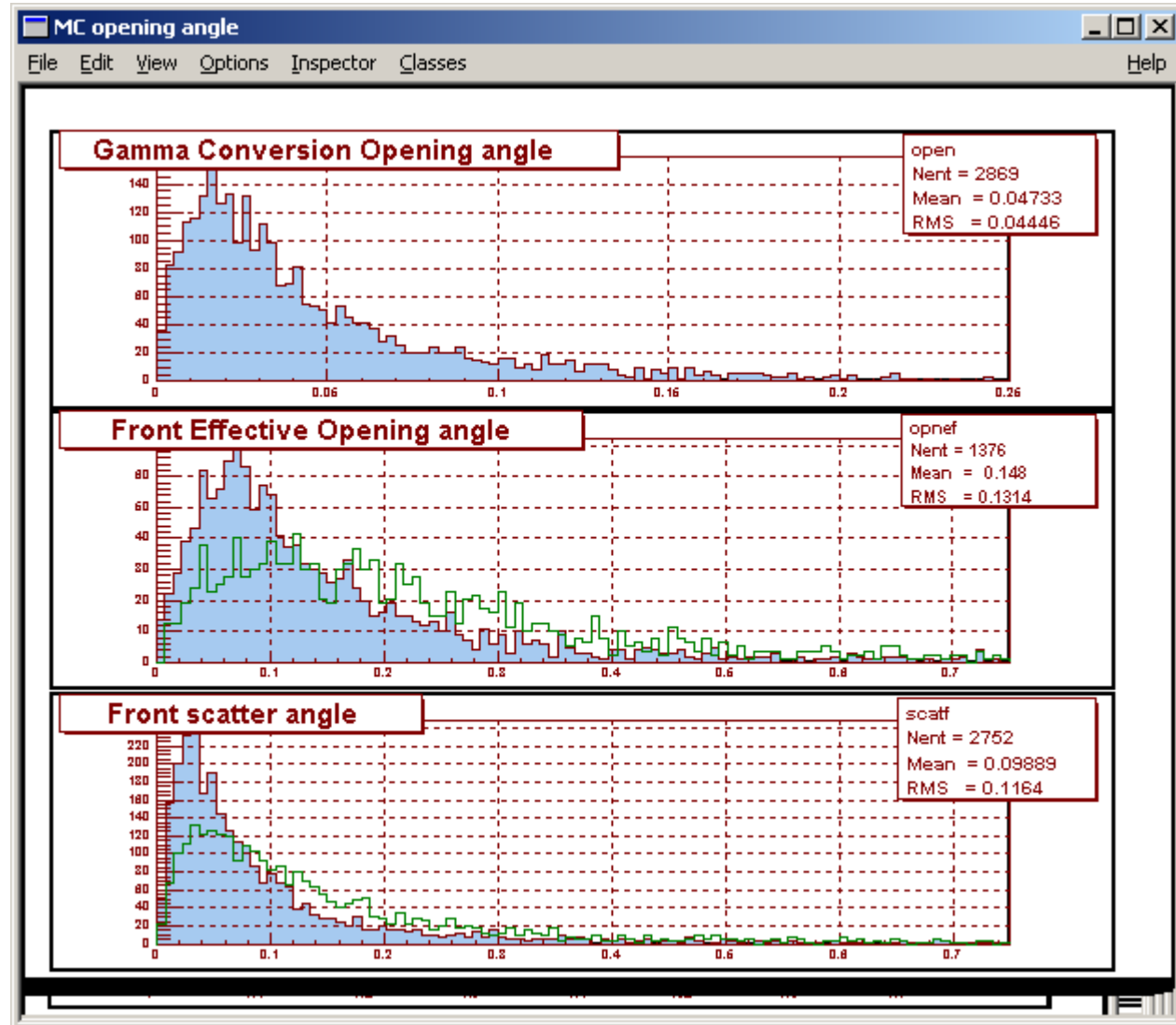
Top row: Opening angle of electron and positron at gamma conversion point

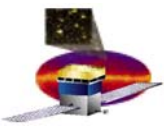
- Before any material effects
- Mean ~47 mr at 100 MeV

Second row: The opening angle seen by the tracker

- Form vector from first two hits in tracker for electron and positron (from McPositionHits)
- Determine angle between these two vectors
- Mean ~15 mr at 100 MeV
- Blue is Front section
- Green is Back section

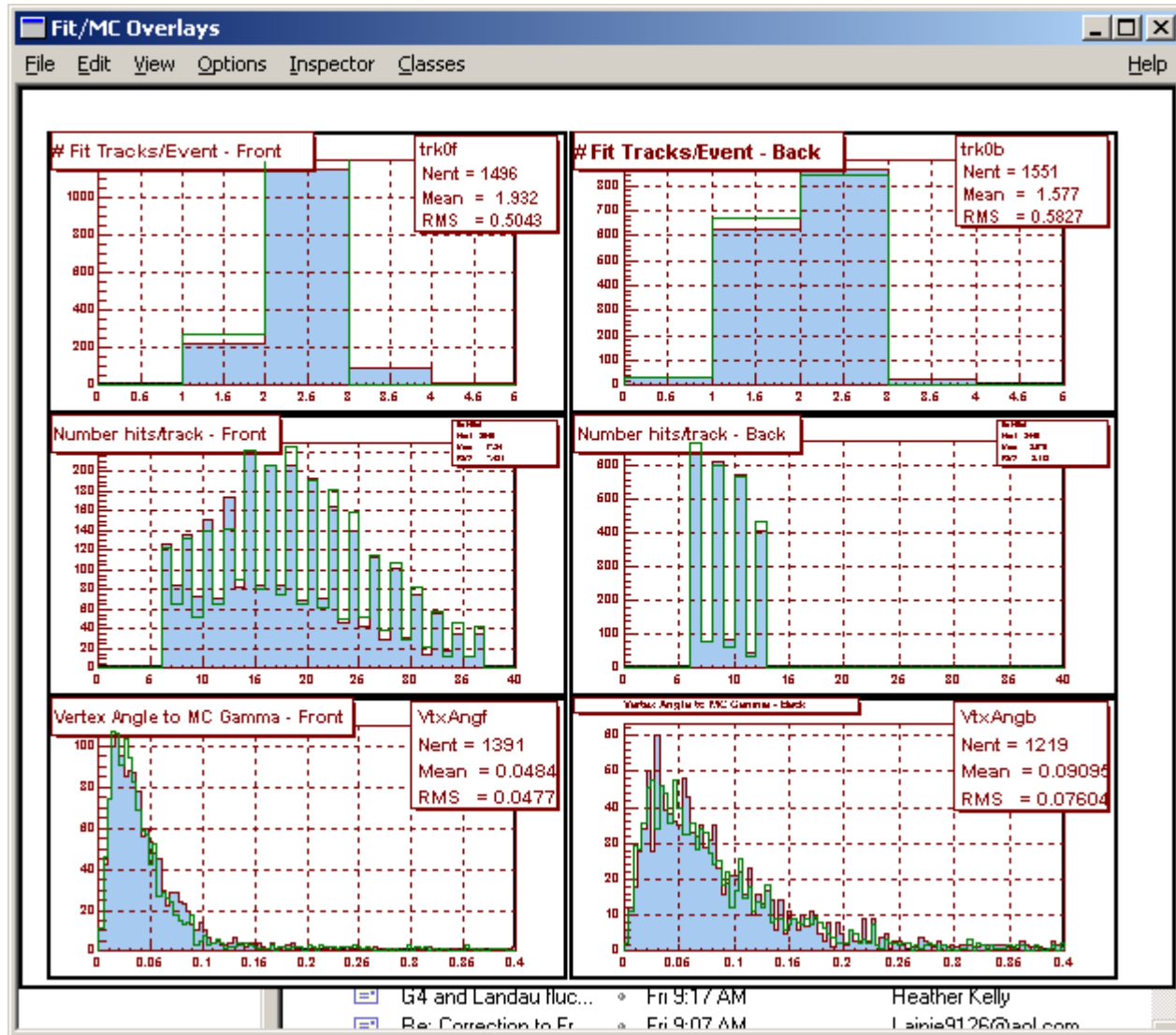
Third Row: Angle between electron (or positron) at creation versus measured vector in Tracker

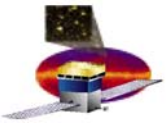




MC / Reconstruction Comparisons

- **MC Gammas**
 - Energy = 100 MeV
 - Downward going
- Left Column is for Front section of tracker
- Right Column is for the Back section
- First row is number of tracks/event
- Second row is number of hits/Track
- Third row is pointing resolution
- Blue filled histograms are reconstruction
- Green lines are MC estimates





Plans/Ideas/Etc.

- **Work in Progress!!!**
 - Not going as fast as I would like either...

- **Plans:**
 - **Continue developing tools to help understand tracking reconstruction issues**
 - Track resolvability
 - Track lengths versus energy
 - Number expected tracks versus energy
 - Number of secondaries expected
 - Etc.

 - **Gamma (versus Track) reconstruction issues**
 - In particular, how above affect resolution
 - What are the most effective cuts?

 - **Come back to questions of "requirements" for track reconstruction**

- **Ideas / suggestions welcome!!**