# Tool and Algorithm to select “Reconstructible” MCParticles and Events

24 / 04 / 2014

# MCRctblTool

- The Gaudi tool **MCRctblTool** provides methods to
  - Qualify as reconstructible an **MCParticle**
  - Count the number of reconstructible **MCParticles** in the event

## Method rctblFromPoints

- Qualifies an MCParticle
  - Check if the **MCParticle** comes from the PV
  - Extrapolate the **MCParticle** to each SciFi layer and see if it is inside
  - Check the hit pattern

## Method rctblFromClusters

- Qualifies an MCParticle
  - Check if the **MCParticle** comes from the PV
  - Get the associated **SciFiClusters**
  - Check the hit pattern

# MCRctblTool

- **Check if the MCParticle comes from the PV**
  - Done with the method **criterionFromPV**
  - Accept only particles with charge = $\pm 1$
  - Accept only particles coming directly from the PV or produced within 1 mm
  - Accept only particles that reach the last SciFi layer without decaying or emitting other particles
  - No switches are available to modify this behavior

- **Check the hit pattern**
  - Done with the method **criterionHitPattern**
  - Count the hits in each of the 4 half-stations
  - Require at least $X$ hits in (nearTop AND farTop) OR (nearBot AND farBot)
    - $X$ can be set through the job option MinLayersPerParticle. Default is 7
  - Require at least $Y$ hits both in the near and in the far half-station
    - $Y$ can be set through the job option MinLayersPerHalfStation. Default is 3
  - In addition, a cut can be applied on the number of hits per station
    - Allows to discard particles passing through the modules' overlap region
    - Job option MaxLayersPerStation. Default is 8

# MCRctblTool

**Note:**

- **rctblFromPoints** needs as input only MCParticles
- **rctblFromClusters** needs as input: MCParticles, SciFiClusters and the links between them
- The performance of the two methods was compared and it was confirmed that they give very similar results
  - More studies in the pipeline ...

## Method countRctblMCP

- Count the number of reconstructible MCParticles in the event
  - Uses either the **rctblFromPoints** or the **rctblFromClusters** method

# RctblEventFilter

- RctblEventFilter is a Gaudi algorithm for selecting “good” events

## Selection criteria

- **Require at most $X$ SciFiClusters (optional)**
  - Can reject busy events. By default this cut is not applied
- **PV**
  - Require the event to contain exactly 1 MC Primary vertex
  - Apply a cut on the $z$ position of the PV
    - Job options PVZMin and PVZMax
- **Number of reconstructible particles**
  - Uses MCRctblTool to count the number of reconstructible particles in the event
  - Uses either the rctblFromPoints or the rctblFromClusters method
    - Job option UseMCAssoc
  - Require at least $X$ reconstructible MCParticles
    - Job option MinRctblTracks

# Usage

- The Gaudi components described in these slides are part of the package **SciFi/MCTools**
  - See **https://svnweb.cern.ch/cern/wsvn/bgv/sw/trunk/SciFi/MCTools**

- Usage of the **MCRctblTool**: see the **RctblEventFilter** algorithm
- Usage of the **RctblEventFilter** algorithm: see the job option file

  **https://svnweb.cern.ch/cern/wsvn/bgv/sw/trunk/SciFi/MCTools/options/RunTestJob.py**
- Some details:
  - The algorithm uses the **setFilterPassed** approach
  - The algorithm should be put inside a **GaudiSequencer**
  - When the algorithm is executed it determines if the event is “good” or not and sets its “FilterPassed” to true or false accordingly
  - If the event was not “good” the algorithms in the **GaudiSequencer** that follow **RctblEventFilter** will not be executed
    - This is the default behavior of **GaudiSequencer**. If needed, it can be modified with job options like **IgnoreFilterPassed** and **ShortCircuit**