Basics: EDM4Hep files (central MC Production)

In the following, we will be using MC files (Delphes) that have been produced as part of the so-called “spring2021” production campaign.

Events files are in EOS:
/eos/experiment/fcc/ee/generation/DelphesEvents/spring2021/IDEA

Name of the collections that are present on these “spring2021” files (the collections that we will use here) :

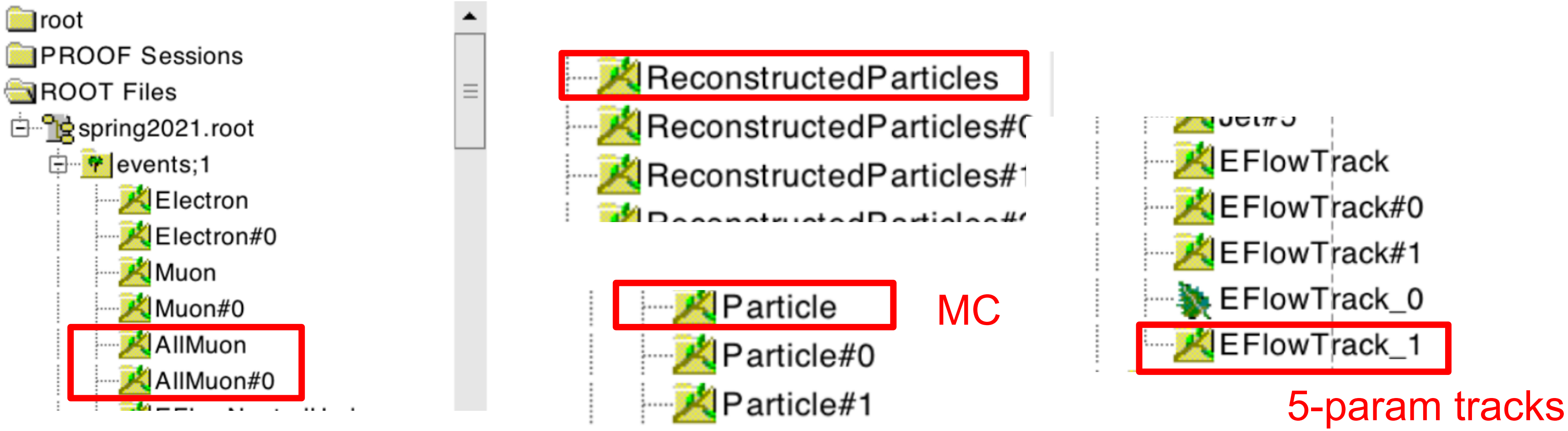
Monte-Carlo particles : Particle
Reconstructed particles : ReconstructedParticles
5-param tracks (i.e. (d0, phi, rho, z0, tanLambda)) : EFlowTrack_1
muons : AllMuons

(The names of the collections are configurable, i.e. files produced with a different configuration may have collections with a different name).

See https://github.com/HEP-FCC/FCCAnalyses/tree/master/examples/basics

Basics: EDM4Hep files (central MC Production)

See https://github.com/HEP-FCC/FCCAnalyses/tree/master/examples/basics



- Branches without a pound (#) in their name: Electron (1), Muon (2), AllMuon (3), EFlowNeutralHadron (4), Particle (5), Photon (6), ReconstructedParticles (7), EFlowPhoton (8), MCRecoAssociations (9), MissingET (10), ParticleIDs (11), Jet (12), EFlowTrack (13), EFlowTrack_1 (14). They refer to collections of objects.
  - NB: "Particle" denotes the collection of Monte-Carlo particles. "Muon" contains the isolated muons, while "AllMuon" contains all muons, isolated or not.
- Branches with a pound in their name: Each of the object collections listed above, e.g. "Collection", has up to six associated collections of references, i.e. indices that point to another or to the same object collection. They are labeled Collection#i, with i = 0 ... 5. For example, the Muon collection has one single associated collection of references, Muon#0.

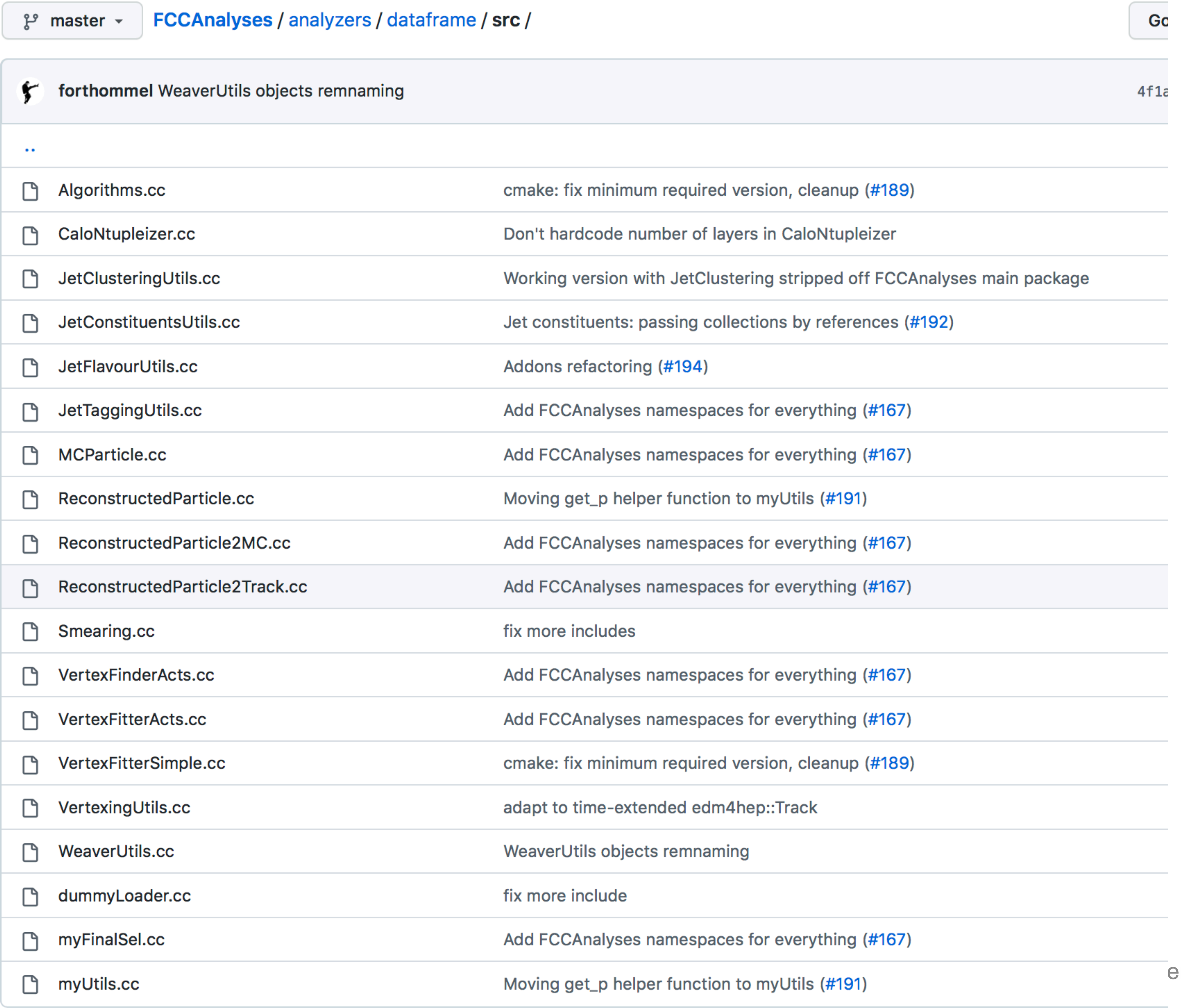
  To figure out which collection is pointed to by Muon#0 (or by any other collection of references), one can look at the value of Muon#0.collectionID (see screenshot below). The collectionID of Muon#0 is the collection number 7 (in the example file used here), which, in the list of "object collections" above, corresponds to the collection of ReconstructedParticles. Indeed, the Muon collection itself contains nothing (see screenshot below): all the information is contained in the ReconstructedParticles. The Muon collection, together with Muon#0, just provides a convenient way to access, among the ReconstructedParticles, those that were identified as muons.

Basics: FCCAnalyses

- See https://github.com/HEP-FCC/FCCAnalyses
  - analysers/dataframe : contains many built-in functions to manipulate the edm4hep collections, and various built-in algorithms
  - Example:
    - MCParticle.h contains code to get the phi, theta angles of MCParticles
    - Many methods coded, use the MCParticle namespace to use them

- Useful analysis information here:
  https://github.com/HEP-FCC/FCCeePhysicsPerformance/tree/master/General

- Oldish talk that describes FCCAnalysis (Clement):
  https://indico.cern.ch/event/982690/contributions/4138504/attachments/2162441/3648904/FCCAnalyses_clement.pdf



master FCCAnalyses / analyzers / dataframe / src /

forthommel WeaverUtils objects remnaming 4f1a

..	
Algorithms.cc	cmake: fix minimum required version, cleanup (#189)
CaloNtupleizer.cc	Don't hardcode number of layers in CaloNtupleizer
JetClusteringUtils.cc	Working version with JetClustering stripped off FCCAnalyses main package
JetConstituentsUtils.cc	Jet constituents: passing collections by references (#192)
JetFlavourUtils.cc	Addons refactoring (#194)
JetTaggingUtils.cc	Add FCCAnalyses namespaces for everything (#167)
MCParticle.cc	Add FCCAnalyses namespaces for everything (#167)
ReconstructedParticle.cc	Moving get_p helper function to myUtils (#191)
ReconstructedParticle2MC.cc	Add FCCAnalyses namespaces for everything (#167)
ReconstructedParticle2Track.cc	Add FCCAnalyses namespaces for everything (#167)
Smearing.cc	fix more includes
VertexFinderActs.cc	Add FCCAnalyses namespaces for everything (#167)
VertexFitterActs.cc	Add FCCAnalyses namespaces for everything (#167)
VertexFitterSimple.cc	cmake: fix minimum required version, cleanup (#189)
VertexingUtils.cc	adapt to time-extended edm4hep::Track
WeaverUtils.cc	WeaverUtils objects remnaming
dummyLoader.cc	fix more include
myFinalSel.cc	Add FCCAnalyses namespaces for everything (#167)
myUtils.cc	Moving get_p helper function to myUtils (#191)

erez

SW tutorial, Vertexing in FCCAnalyses

This tutorial will teach you how to:

- fit some tracks to a common vertex in FCCAnalyses,
- recontruct the primary vertex and the primary tracks
- retrieve the tracks corresponding to a specific flavour decay in FCCAnalyses
- produce flat ntuples with observables of interest with FCCAnalyses
- build your own algorithm inside FCCAnalyses

It consists in two parts:

- Run basic vertex fits in FCCAnalyses
  - May also use these examples to explain basic things on FCCAnalyses if needed
- Setting up building blocks for a $\tau \rightarrow 3\mu$ analysis
  - More advanced

Based on (existing) Delphes files in the edm4hep format.