

# Updates in edm4hep and plans for Monte-Carlo productions

Jan Eysermans (MIT), Juraj Smiesko (CERN)

Physics Performance meeting – March 18 2024

# Pointers and Contacts



## FCC Analysis community

**Repository**      <https://github.com/HEP-FCC/FCCAnalyses/tree/master>

**Forum**            <https://fccsw-forum.web.cern.ch/c/fccanalysis/9>

**Egroup**            [FCC-PED-SoftwareAndComputing-Analysis@cern.ch](mailto:FCC-PED-SoftwareAndComputing-Analysis@cern.ch)

## Meetings

- Bi-weekly informal meeting, more kind of a chat
- Next meeting Wednesday March 27, 11 AM, <https://indico.cern.ch/event/1392211/>

# MC Campaigns Overview



## The default campaign is **Winter 2023**

- Used by most analysis (if not all)
- List of samples can be found [here](#)
- Recent production of samples at 365 GeV, see tomorrow Higgs/Top meeting for more details
- Private samples → encourage in porting to central production (contact us)

## Campaign **Spring 2021**

- Who is (still) using it?
- Will be deprecated soon, should move to Winter 2023
- Contact us if you need help to move from Spring 2021 to Winter 2023

## **The MC production and FCCAnalyses software are inherently connected to the key4hep software stack**

- Underneath is the [edm4hep event data model](#), that evolves according to the experience and feedback of the users
  - It's still in beta version
- FCCAnalyses typically uses the latest stable key4hep stack version to be up-to-date with the recent developments in key4hep and auxiliary software (e.g. ROOT)
- MC production uses a fixed key4hep stack in order to generate all the MC under the same conditions
- Conflicts might arise between FCCAnalyses and the MC production

## **Within a few weeks, the first official edm4hep will be released (v1.0)**

- This will break the compatibility with the older edm4hep releases (and so the MC campaign)
- Backwards compatibility from this release onwards guaranteed using Schema Evolution (see details later)

## **We will present here the changes on the edm4hep side and how to deal with our MC/FCCAnalyses**

- First an overview of the edm4hep changes will be given, relevant for the analyzers
- Strategy for current MC campaign and future ones

**Given the expected divergences stack/edm4hep/FCCAnalyses, we have the following strategy in mind**

## **1. Keep support for Winter 2023 campaign**

- At least people can continue their analysis till the FSR is completed
- FCCAnalyses developments with frozen stack version and FCCAnalyses version

## **2. In parallel, we will setup and validate the newest edm4hep v1.0**

- First round of validation and necessary adaptations to FCCAnalyses (now-April)
  - New version of FCCAnalyses – developments in old FCCAnalyses will be ported to the new release
- Gradual reprocessing of entire Winter 2023 campaign to Spring 2024 (April-May-June)
  - Add new features but NO change in detector or machine parameters (i.e. physics stays the same)

**From June onwards, analyzers can switch to the new campaign if desired**

- Can prioritize sample production depending on the needs
- Keeping winter 2023 means no change for analyzers, but not all developments in the new stack and FCCAnalyses will/can be backported (discussion per case per case)

# **Overview of the new edm4hep features and impact on FCCAnalyses**

# EDM4hep 1.0

- *The EDM4hep will reach version 1.0 soon*
  - Introduces fixes but also breaking changes
- Some of the changes/fixes introduced:
  - [Interfaces](#)
  - [ReconstructedParticle: type→PDG](#)
  - [Reverse the direction of the ParticleID relation\(s\)](#)
  - [Vector of weights in EventHeader](#)
- From version 1.0 schema evolution should handle changes in the Datamodel
- Pre EDM4hep 1.0 preserved in Key4hep stack:
  - `source /cvmfs/sw.hsf.org/key4hep/setup.sh -r 2024-03-10`
- And also in FCCAnalyses version: `v0.9.0` and branch: `pre-edm4hep1`
  - Development for pre EDM4hep 1.0 will be cherry picked to master

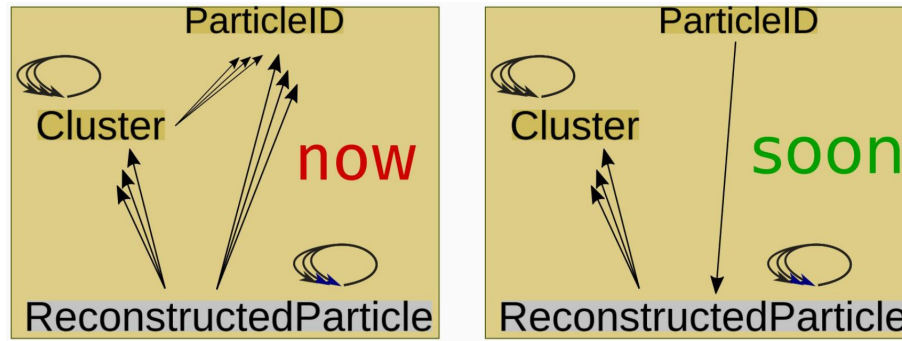
# EDM4hep: Interfaces

- *Creates common interface for similar objects*
- Example:
  - `TrackerHit` ([link](#)) is interface for: `TrackerHit3D` and `TrackerHitPlane`
  - Common members:
    - `cellID` ID of the sensor that created this hit
    - `type` type of the raw data hit
    - `quality` quality bit flag of the hit
    - `time [ns]` time of the hit
    - `eDep [GeV]` energy deposited on the hit
    - `eDepError [GeV]` error measured on eDep
    - `position [mm]`
  - **Notice:** missing `u`, `v` measurement direction vectors
- More information: [Thomas' talk](#)



# EDM4hep: ParticleID adjustments

- *Necessary to have EDM4hep multithreaded*
- Big conceptual change requiring workflow investigation
  - Comments welcome in the [EDM4hep PR](#)
- Simple looping still easy
  - loop through ParticleID collection instead of RecoParticle
- Operations requiring going from RecoParticle back to ParticleID more complex
- More information: [Thomas' talk](#)



# EDM4hep: Event weight vector

- *Allows storing multiple event weights per event*
- EDM4hep `EventHeader` ([link](#)) has now two members:
  - `weight` - global weight of the event
  - `weights` - vector of weights
- The event weight names are stored separately in the file level metadata
- Name of the metadata is standardized in [edm4hep/Constants.h](#) as `edm4hep::EventWeights` constant
- Current value is “EventWeightNames”

# Pre-generated samples website

- *FCC Physics Events website overhauled*
  - <https://fcc-physics-events.web.cern.ch/fcc-physics-events/FCCee/>
  - Uses Bootstrap v5
  - Will be needed for Dirac samples
- Four tag categories:
  - Accelerator
  - Campaign
  - Event type
  - Detector

Example: [FCC-ee | Winter 2023 | Delphes | IDEA](#) samples

# Event Filter

- *Allows more complex filtering rules when generating samples with Gaudi*
- Filter rule can act on any information in MCParticle collection
- The C++ filter rule is JIT compiled
- Can be provided as a string or as cxx/hxx file
- Example ([k4Gen PR](#)):
  - `#include "edm4hep/MCParticleCollection.h"`

```
bool filterRule(const edm4hep::MCParticleCollection* inColl) {  
    return inColl->size() > 1000;  
}
```