

# Vertex EDM

Joe Osborn, BNL  
June 8, 2023

# Current Vertex Object

```
## =====  
## Vertexing  
## =====  
  
edm4eic::Vertex:  
  Description: "EIC vertex"  
  Author: "W. Armstrong, S. Joosten, based off EDM4hep"  
  Members:  
    - int32_t          primary      // Boolean flag, if vertex is the primary vertex of the event  
    - float            chi2         // Chi-squared of the vertex fit  
    - float            probability  // Probability of the vertex fit  
    - edm4hep::Vector3f position    // [mm] position of the vertex.  
    ## this is named "covMatrix" in EDM4hep, renamed for consistency with the rest of edm4eic  
    - edm4eic::Cov3f   positionError // Covariance matrix of the position  
    - int32_t          algorithmType // Type code for the algorithm that has been used to create the vertex - check/set the colle  
    ## Additional parameter not in EDM4hep: vertex time  
    - float            time         // Vertex time  
  VectorMembers:  
    - float            parameters   // Additional parameters related to this vertex - check/set the collection parameter "Vertex  
  OneToOneRelations:  
    ## @TODO: why one and not multiple particles?  
    - edm4eic::ReconstructedParticle associatedParticle // reconstructed particle associated to this vertex.
```

- Current vertex object in edm4eic that is created at end of vertexing algorithm IterativeVertexFinder
- Missing some notable pieces of data, and in any case does not match the Acts vertex EDM well

# Current Vertex Object

```
## =====  
## Vertexing  
## =====  
  
edm4eic::Vertex:  
  Description: "EIC vertex"  
  Author: "W. Armstrong, S. Joosten, based off EDM4hep"  
  Members:  
    - int32_t          primary      // Boolean flag, if vertex is the primary vertex of the event  
    - float            chi2         // Chi-squared of the vertex fit  
    - float            probability  // Probability of the vertex fit  
    - edm4hep::Vector3f position    // [mm] position of the vertex.  
    ## this is named "covMatrix" in EDM4hep, renamed for consistency with the rest of edm4eic  
    - edm4eic::Cov3f   positionError // Covariance matrix of the position  
    - int32_t          algorithmType // Type code for the algorithm that has been used to create the vertex - check/set the colle  
    ## Additional parameter not in EDM4hep: vertex time  
    - float            time         // Vertex time  
  VectorMembers:  
    - float            parameters   // Additional parameters related to this vertex - check/set the collection parameter "Vertex  
  OneToOneRelations:  
    ## @TODO: why one and not multiple particles?  
    - edm4eic::ReconstructedParticle associatedParticle // reconstructed particle associated to this vertex.
```

- What is “probability of the vertex fit?” Acts returns no such object, unless this is referring to the NDF of the fit (this is currently stored in this object)
- No uncertainty on the time - why not just make this a 4D vector with a complete 4D covariance matrix?
- No way to associate which reconstructed tracks belong to a vertex
- Do we need an “algorithmType” and 32 bit integer for a boolean noting whether or not the vertex is primary?



# Proposal

## edm4eic::Vertex

```
## =====
## Vertexing
## =====

edm4eic::Vertex:
  Description: "EIC vertex"
  Author: "J. Osborn"
  Members:
    - bool          primary          // Boolean flag, if vertex is the primary vertex of the event
    - float         chi2             // Chi-squared of the vertex fit
    - float         ndf             // NDF of the vertex fit
    - edm4hep::Vector4f fullPosition // [mm] position + time t0 of the vertex.
    ## this is named "covMatrix" in EDM4hep, renamed for consistency with the rest of edm4eic
    - edm4eic::Cov4f fullPositionError // Covariance matrix of the position
  VectorMembers:
    - float         parameters      // Additional parameters related to this vertex - check/set the collection
    " for the parameters meaning.
  OneToManyRelations:
    - edm4eic::ReconstructedParticle associatedParticles // reconstructed particles associated to this vertex.
```

## Acts::Vertex

```
private:
  Vector4 m_position = Vector4::Zero();
  SymMatrix4 m_covariance = SymMatrix4::Zero();
  std::vector<TrackAtVertex<input_track_t>> m_tracksAtVertex;
  double m_chiSquared = 0.; // chi2 of the fit
  double m_numberDoF = 0.; // number of degrees of freedom
```

- Vertex object should contain:
  - 4 vector (position,  $t_0$ )
  - Full 4D covariance to include timing information
  - $\chi^2$  and NDF
  - Pointers to the reconstructed tracks associated to a vertex (or some way to relate tracks to vertex)
- What else?

# Proposal

- We also need to consider how truth vertices will be handled, especially with the implementation of background
- As of now there is no way to directly evaluate the vertexing performance other than comparing MCParticle vertices to the reconstructed vertex
  - This will be very cumbersome soon and is not a long term solution
- Do we need a truth vertex object? Would it be useful for keeping track of backgrounds?
  - e.g. truth vertex might have a flag for what type of event it is associated to (primary, beam-gas, synchrotron, something else...
  - Truth vertex inherently needs less information, so would offer some memory savings (e.g. truth vertex does not have covariance,  $\chi^2$ , etc.

# Thoughts?