
Command line tools for Podio

DESY Summer Student Programme, 2021

Juan Manuel Moreno Pérez

National University of Colombia

Supervisor
Gernot Maier



September 9, 2021

Abstract

Podio is an EDM tool that generates all code from YAML descriptions. As part of the Key4Hep project it is necessary that it has some tools that are already available in the iLC software, in this file we present advances made during the summer student project implementing tools similar to anajob and dumpevent.



Contents

1	Introduction	1
2	Development	1
3	Results	3
4	Conclusions	3

1 Introduction

The Key4hep project is a software construction project to make a common framework for all future HEP experiments to use, the purpose of these is to implement all usual and necessary libraries used in anything from simulation to analysis. These was motivated by a common software developed by ILC and CLIC. The intention is that these facilitates the usage and implementation of new packages ensuring compatibility with already existing ones like ROOT, Geant4, DD4hep, Gaudi and podio which will be the main focus of the project.

The podio package is an EDM package that generates code based on YAML description, it is used within the Key4hep project, it can be used to generate arbitrary data models which come great when using different backend I/O. // LCIO is the event data model used currently by the iLC software, it has been developing for 15 years and due to that there are multiple tools developed for this, we focus on two of them anajob and dumpevent which are used to take a look at the files without making any code for that. Anajob and dumpevent are really similar yet very different, the anajob gives us an overview of the kind of data contained in the event but only prints the data types in the event while the dumpevent prints the specific values contained on these events, this is why is important to begin by the development of the anajob first.

2 Development

To develop the tool we want to make that gives as an overview like the anajob tool it is necessary to look at the output produced by anajob which is presented here:

```

////////////////////////////////////
EVENT: 0
RUN: 0
DETECTOR: ILD_l5_v02
COLLECTIONS: (see below)
////////////////////////////////////

-----
COLLECTION NAME          COLLECTION TYPE          NUMBER OF ELEMENTS
-----
BeamCalCollection       SimCalorimeterHit       2
ECalBarrelScHitsEven    SimCalorimeterHit       4870
ECalBarrelScHitsOdd     SimCalorimeterHit       4878
ECalBarrelSiHitsEven    SimCalorimeterHit       2334
ECalBarrelSiHitsOdd     SimCalorimeterHit       2371
ECalEndcapScHitsEven    SimCalorimeterHit       3901
ECalEndcapScHitsOdd     SimCalorimeterHit       3896
ECalEndcapSiHitsEven    SimCalorimeterHit       2283
ECalEndcapSiHitsOdd     SimCalorimeterHit       2100
ECalEndcapRingCollection SimCalorimeterHit       411
FTDCollection           SimTrackerHit            265
HCalBarrelRPCHits       SimCalorimeterHit       649
HCalECRingRPCHits       SimCalorimeterHit       17
HCalEndcapRPCHits       SimCalorimeterHit       1794
HCalBarrelRegCollection SimCalorimeterHit       6536
HCalEndcapRingCollection SimCalorimeterHit       211
HCalEndcapsCollection   SimCalorimeterHit       14116
LHCalCollection         SimCalorimeterHit       116
LumiCalCollection       SimCalorimeterHit       54
MCParticle              MCParticle               904
SETCollection           SimTrackerHit            127
SITCollection           SimTrackerHit            490
TPCCollection           SimTrackerHit            18999
TPCLowPtCollection      SimTrackerHit            0
TPCSpacePointCollection SimTrackerHit            0
VXDCollection           SimTrackerHit            431
YokeBarrelCollection    SimCalorimeterHit       50
YokeEndcapsCollection   SimCalorimeterHit       464
-----

```

Figure 1: Output produced by anajob on an example file

In the next images some changes made to the current podio state will be presented, however as this is a small report to present more the results than the process the more

technical details will be left out. For further information this is the link to the repo on github: <https://github.com/jummorenope/podio.//>

The first step in order to be able to read any file was implementing a function that could read the two type of files that are currently supported by podio which are SIO files and ROOT files.

```

1  #include "ROOTReader.h"
2  #include "SIOReader.h"
3
4  std::unique_ptr<podio::IReader> getReader(const std::string& FileName){
5      if(FileName.substr(FileName.length()-4)=="root"){
6          return std::make_unique<podio::ROOTReader>();
7      }
8      else{
9          return std::make_unique<podio::SIOReader>();
10     }
11 }

```

The next step after having done this was implementing some functions within the readers to make sure they were able to circle through events, with this being implemented it was possible to create a loop able to circle through the events required like this.

```

1  //Printing important info of the file
2  std::cout<<"FileName: "<<FileName<<std::endl;
3  std::cout<<"Number of events: "<<eventNumber<<std::endl;
4  std::cout<<std::endl;
5
6  //Getting Table containing the info about the collections
7  const auto collIDTable = reader->getCollectionIDTable();
8  const auto collNames = collIDTable->names();
9  //Iterating over all events to get size for each event
10 for(int i=startEvent; i<readEvent; i++){
11
12     std::cout<<"Event Number " <<i<<std::endl;
13     std::cout<<std::left<<std::setw(30)<<"Name"<<std::left<<std::setw(40)
14     <<"Type"<<"Collection Size"<<std::endl;
15     std::cout<<"-----"
16     <<std::endl;
17
18     reader->readEvent();
19     //Iterating over the collections to get each kind of data type
20     for (const auto& name : collNames) {
21
22         const auto coll = reader->readCollection(name);
23         auto size = coll->size();
24         const auto type = coll->getValueTypeName();
25         //Printing Collection Names
26         std::cout<<std::left<<std::setw(30)<<name<<std::left<<std::setw(40)

```

```

27     <<type<<size<<"\n";
28 }
29 reader->endOfEvent();
30 std::cout<<std::endl;
31 }

```

3 Results

With the previous code this output can be produced:

```

FileName: example.root
Number of events: 2000

Event Number 0
Name          Type          Collection Size
-----
info          EventInfo     1
mcparticles  ExampleMC    10
moreMCs      ExampleMC    10
mcParticleRefs ExampleMC     0
hits         ExampleHit    2
clusters     ExampleCluster 3
refs         ExampleReferencingType 2
refs2        ExampleReferencingType 1
Component    ExampleWithComponent 1
OneRelation  ExampleWithOneRelation 2
WithVectorMember ExampleWithVectorMember 2
WithNamespaceMember ex42::ExampleWithNamespace 25
WithNamespaceRelation ex42::ExampleWithARelation 5
WithNamespaceRelationCopy ex42::ExampleWithARelation 5
strings      ExampleWithString 1
arrays       ExampleWithArray 1
fixedWidthInts ExampleWithFixedWidthIntegers 3

```

Figure 2: Caption

This is actually pretty similar to the one found in anajob 1 with some small differences in format which are not really important, the dumpevent has not yet been developed but there is already a plan to add some functions in the CollectionBase dependency at podio.

4 Conclusions

- Podio is the EDM tool for the key4hep project and it supports multiple I/O backends.
- An equivalent to anajob has been developed, it required a few changes to the code itself.
- Both the current and the old code had to be working.
- An equivalent to dumpevent started to be developed but faced some issues due to the nature of podio.

This challenges will still be worked upon.

Acknowledgements

To Thomas and Frank who gave me every possible assistance during the Summer Student internship.