HERDOS New User Quick Guide

Z. Tang

IHEP

Access to the software

to setup cvmfs, please following the instruction at
https://herd.ihep.ac.cn/internal/herdos/manual/installation/installation.html
and contact Xiaowei JIANG (jiangxw@ihep.ac.cn) if any problem

- cvmfs:
  - /cvmfs/herd.ihep.ac.cn/HERDOS/el9_amd64_gcc11/Release/v00-10
  - /cvmfs/herd.ihep.ac.cn/HERDOS/centos7_amd64_gcc850/Release/v00-10
  - /cvmfs/herd.ihep.ac.cn/HERDOS/centos7_arm64_gcc850/Release/v00-10

```
├── ExternalLibs
│   ├── bashrc.sh
│   ├── geant4
...
├── offline
│   ├── install
│   └── installation
└── setup.sh
```

source this file if build your own HERDOS

source this file if use the official HERDOS build

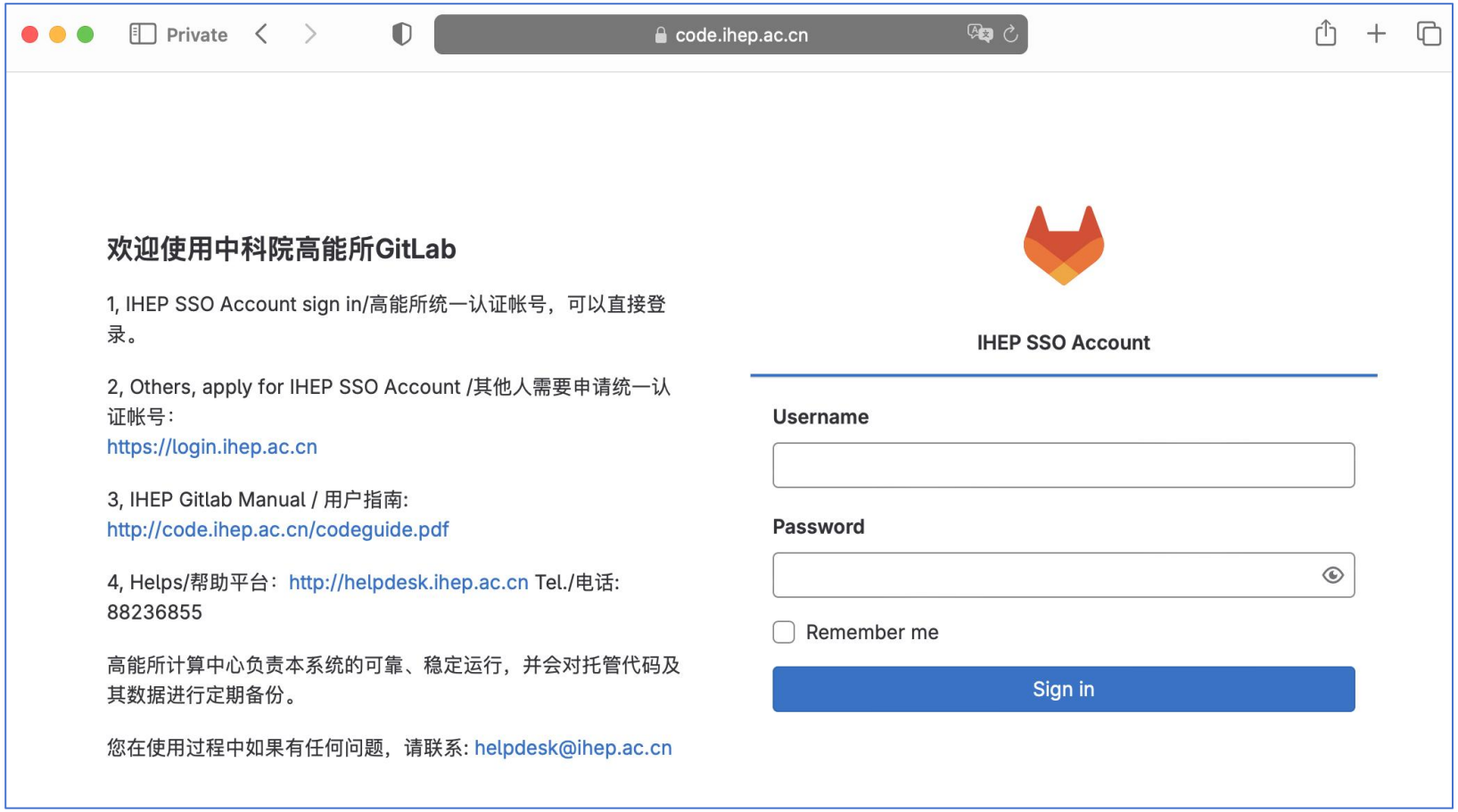
Without cvmfs, please following the instruction at
https://herd.ihep.ac.cn/internal/herdos/manual/installation/installation.html to build your own external libs.
please contact Teng LI (tengli@sdu.edu.cn) if any problem in building external libs

Access to code repository

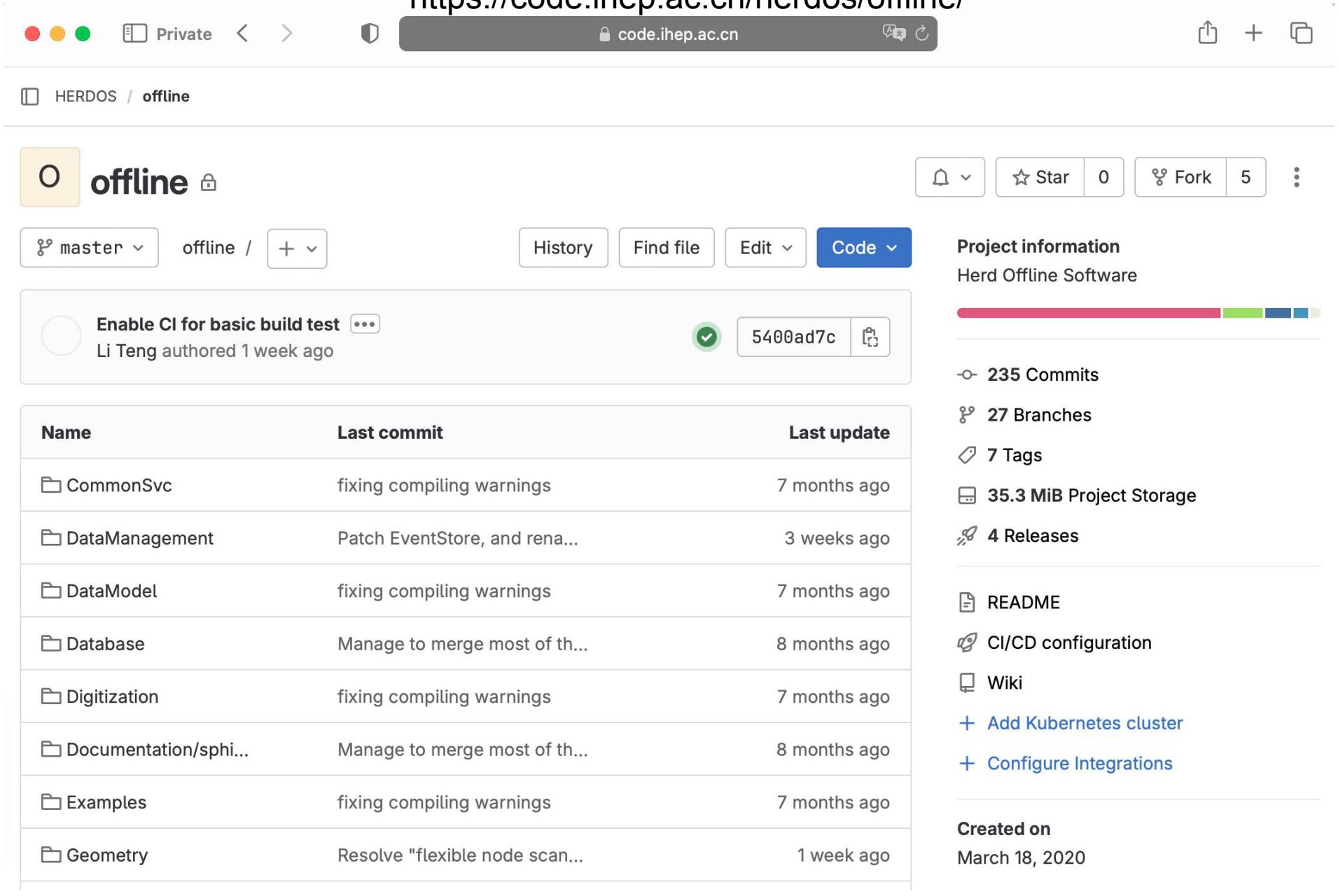
- Gitlab server: https://code.ihep.ac.cn

1. Login with IHEP SSO
   - Apply on https://login.ihep.ac.cn
2. Access to repository should be apply in advance by email to Zhicheng TANG (tangzhch@ihep.ac.cn) from group contact person

- can be applied in one step by sending list of emails to Zuhao LI (lizh@ihep.ac.cn) from group contact person



https://code.ihep.ac.cn/herdos/offline/



Building HERDOS

- Configure your working environment, including ssh key, sourcing external library environment, etc
- clone the git repository to your working directory:
  - `git clone git@code.ihep.ac.cn:herdos/offline.git`
- build with provided shell script
  - `cd offline`
  - `./build.sh` (or other versions)
- setup the environment for your own build
  - `source install/setup.sh` (this step is needed for further steps)

Running MC Simulation & Digitization

- `python3 $HERDOS_INSTALL/scripts/SimConfiger/devrun.py --particle proton --energy 100 --geometry v2022a/v2022a-f.xml --g4mac iso-R1.8m -N 10000 -o demoJuly26.root --fitdigi`

geant4 particles names or ion names like z2, z3, ...

in GeV. can be two numbers for a range. can also use `--Ekn`

include digitization in the run. also `scddigi`, `calodigi`, ...

name of the geometry file, under $HERDOS_INSTALL/compact/

name of the genaration config (geant4 macro file), under $HERDOS_INSTALL/g4mac/, can be multiple names

```
usage: devrun.py [-h] [-N NEVT] [-g GEOMETRY] [-m G4MAC [G4MAC ...]] [-s SEED] [--physicslist PHYSICSLIST]
                 [--energy ENERGY [ENERGY ...]] [--Ekn EKN] [--particle PARTICLE] [--runid RUNID] [--withss]
                 [--withbase] [--solarpanel a1_0 a1_1 a1_2 a2_0 a2_1 a2_2 a3_0 a3_1] [--compact] [--psddigi]
                 [--fitdigi] [--scddigi] [--calodigi] [--dryrun] [-o OUTPUT] [--run RUN] [--info INFO]
                 [--verbose {TEST,INFO,DEBUG,WARN,WARNING,ERROR,FATAL}]

optional arguments:
  -h, --help            show this help message and exit
  --withss              include full space station in simulation (default: off)
  --withbase            include basebox in simulation (default: off)
  --solarpanel a1_0 a1_1 a1_2 a2_0 a2_1 a2_2 a3_0 a3_1
                        solar panel angle parameters in deg (default: all 0)
  --compact             compact output file (remove some variables)
  --psddigi             enable PSD digi
  --fitdigi             enable FIT digi
  --scddigi             enable SCD digi
  --calodigi            enable CALO digi
  --dryrun              test args without running
  -o OUTPUT, --output OUTPUT
                        the output file
  --run RUN             set the run id of this job between 0 and 2147483647 (uint32)
  --info INFO           a short message, to be put in the file title
  --verbose {TEST,INFO,DEBUG,WARN,WARNING,ERROR,FATAL}
                        limit global log level to at least this value

Standard simulation options:
  -N NEVT, --nevt NEVT  Number of events to simulate (default: 1)
  -g GEOMETRY, --geometry GEOMETRY
                        select a different dd4hep geometry xml file
  -m G4MAC [G4MAC ...], --g4mac G4MAC [G4MAC ...]
                        the macro file to run for simulation. NOTE: if OUTPUT is set to vis then this file is used
                        as vis macro
  -s SEED, --seed SEED  the initial random seed, or 0 to auto generate from urandom (default: 8823)
  --physicslist PHYSICSLIST
                        One of the predefined physics list to use
  --energy ENERGY [ENERGY ...], -E ENERGY [ENERGY ...]
                        set mono energy or energy range [GeV] in G4GPS
  --Ekn EKN, --ekn EKN  set mono energy per nucleon [GeV] in G4GPS, only works if also specify --particle
  --particle PARTICLE, -p PARTICLE
                        set primary particle type in G4GPS
  --runid RUNID         set the runID of this sim job [0, 2147483647] (uint32)

Other regular arguments are treated as input data files to be processed
```

- Digitization can be run separated, e.g.

```
python3 $HERDOS_INSTALL/scripts/GlobalTrack/run_recon.py --calodigi --input demoJuly26.root --output digi.root
```

please contact Zhicheng TANG if any problem on simulation(tangzhch@ihep.ac.cn)

FInd details also in user manual

Running Reconstruction

Reconstruction can be run separately or all together, e.g.

```
python3
$HERDOS_INSTALL/scripts/GlobalTrack/run_rec
on.py
--calopca
--fitcluster
--fittrack
--scdcluster
--scdtrack
--globaltrack
--input digi.root
--output reco.root
```

A new interface to access all functions in one script is under development

- Currently there are sevaral madurate reconstruction available:
  - CaloPCA
  - FITCluster
  - FITTrack
  - SCDCluster
  - SCDTrack
  - GlobalTrack
- Other algorithms under development, see Ming XU's presentation for details
- Contribution to the algorithm development is welcome

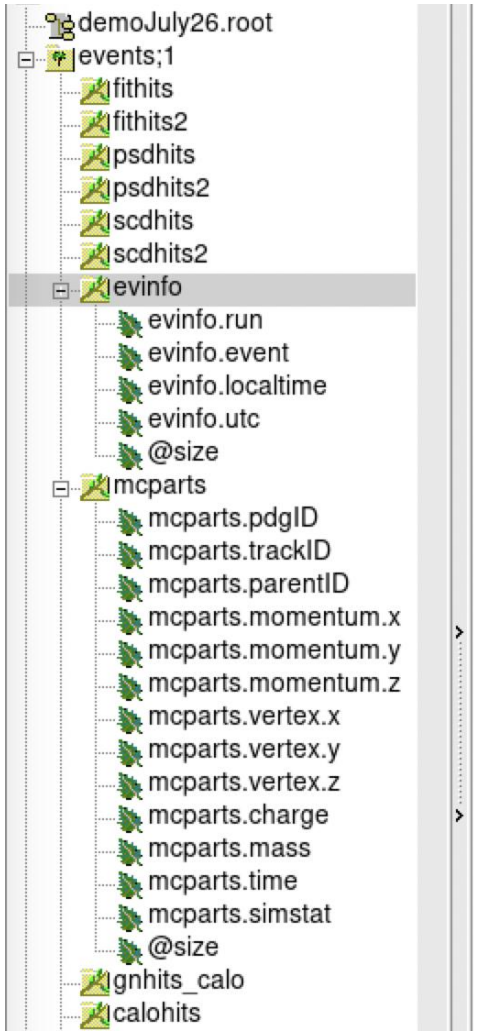
Analyzing of Data File

- The HERDOS EDM is based on podio, all data are organized in form of “collections” in each event
- The generated output is a self defined format root file, in the “events” tree, with generated collections as branches
- Several ways to read the root file
  - inside HERDOS
  - with podio (EventStore)
  - simple way (TTreeReader)
  - ...
- The available collections depends on the running algorithms, e.g.:

simulation	digization	reconstruction
evinfo	calodigi	caloPCA
mcparts	scddigi	scdKalmanTracks
calohits	psddigi	scdLinFitTracks
gnhist_calo	fitdigi	scdclusters
fithits	trddigi	globaltracks
scdhits	...	fitrclusters
psdhits		fittrack
...		...

https://code.ihep.ac.cn/herdos/offline/-/blob/master/DataModel/EventDataModel/datalayout.yaml

```
73
74    edm::Event:
75       Description: "event info"
76       Author: "Z.Tang"
77       Members :
78          - int run       // run id
79          - int event     // event id in the run
80          - int localtime // reserved for DAQ compute
81          - double utc    // reserved for UTC time fr
82 ## %ENDCODE%</sticky>
83
84 ## ---++++ MC Particles
85 ##
86 ## Infomation about simulated particles
87 ##    * pdgid, trackid, parentid (int)
88 ##    * position (float*3)
89 ##    * momentum (float*3)
90 ##    * time (float) (charge, mass, ...) (short)
91 ##
92 ## yaml code:
93 ## <sticky>%CODE{"c"}%####
94
95    edm::MCParticle:
96       Description: "MC particle"
97       Author: "Z.Tang"
98       Members:
99          - int pdgID                    // PDG code
100         - int trackID                  // index of
101         - int parentID                 // index of
102         - edm::Vector3f momentum       // particle
103         - edm::Vector3f vertex         // productic
104         - short charge                 // atomic ch
105         - uint16_t mass                // atomic ma
106         - float time                   // creation
107         - uint32_t simstat             // (opt) sta
108 ## %ENDCODE%</sticky>
109
```



including event info, MC truth, simulation hits, digitization hits, trigger info, reconstruction objects,

Demo Code: inside HERDOS

```
46  bool AnalysisAlg::execute()
47  {
48     LogDebug << "Processing event " << mEvt << std::endl;
49     ++mEvt;
50
51     kTrackSimHits = getROColl(TrackingSimHitCollection, "scdhits");
52     kCaloSimHits = getROColl(CaloSimCellCollection, "calohits");
```

read the collection of current event

```
59     if (kCaloSimHits)
60     {
61        for (size_t i=0; i<kCaloSimHits->size(); ++i)
62        {
63           auto edep = kCaloSimHits->at(i).getEdep();
64           mHistEdep->Fill(edep);
65        }
66     }
```

loop the objects in the collection

https://code.ihep.ac.cn/herdos/offline/-/blob/34-enrich-examples/Examples/AnalysisExample/src/AnalysisAlg.cc

```
14  import PodioSvc
15  Isvc = task.createSvc("PodioInputSvc/InputSvc")
16  Isvc.property("InputFile").set("simhits.root")
17
```

The input file is specified in steering python script
(e.g. https://code.ihep.ac.cn/herdos/offline/-/blob/34-enrich-examples/Examples/AnalysisExample/scripts/AnalysisAlg.py)

Instruction on details will be presented in coming presentations from Teng LI

Demo Code: Read with podio

UsePodio.C

```
 1 #include <iostream>
 2 #include "TH1.h"
 3 #include "TCanvas.h"
 4 #include <podio/EventStore.h>
 5 #include <podio/ROOTReader.h>
 6 #include <EventDataModel/CaloSimCellCollection.h>
 7 #include <EventDataModel/MCParticleCollection.h>
 8
 9 using namespace std;
10 using namespace edm;
11
12 void UsePodio(TString filename)
13 {
14    podio::EventStore store;
15    podio::ROOTReader reader;
16    reader.openFile(string(filename));
17    store.setReader(&reader);
18
19
20    TH1F* h = new TH1F("h","h",100,0,120);
21    for (int jentry=0;jentry<reader.getEntries();++jentry)
22    {
23       auto& calohits = store.get<edm::CaloSimCellCollection>("calohits");
24       auto& mcparts = store.get<edm::MCParticleCollection>("mcparts");
25
26       float Edep=0;
27       for (int i=0;i<calohits.size();++i)
28       {
29          edm::CaloSimCell calohit = calohits.at(i);
30          float E = calohit.getEdep();
31          short ix = calohit.getIx();
32          short iy = calohit.getIy();
33          short iz = calohit.getIz();
34          Edep += E;
35       }
36       h->Fill(Edep);
37
38       reader.endOfEvent();
39       store.clear();
40    }
41    TCanvas* c = new TCanvas("c","c",800,600);
42    h->Draw();
43    gPad->SetLogy();
44    c->SaveAs("edep.png");
45 }
```

can be found at https://code.ihep.ac.cn/herdos/offline/-/tree/34-enrich-examples/Examples/StandAloneAnalysis/

Running podio demo code

another script (**`loader.C`**) needed to running this demo

```
1 {
2    gInterpreter->AddIncludePath(gSystem->ExpandPathName("$HERD_EXTLIB_po
  dio_HOME/include"));
3    gSystem->Load("libpodio");
4    gSystem->Load("libpodioRootIO");
5    gSystem->Load("libEventDataModel");
6    gSystem->Load("libEventDataModelDict");
7 }
```

usage: **`root loader.C UsePodio.C -- '"input.root"'`**
or
`root loader.C UsePodio.C'("input.root")'`

A makefile is also provided to compile to executable

```
4 #include <podio/EventStore.h>
5 #include <podio/ROOTReader.h>
```

podio related headers

```
6 #include <EventDataModel/CaloSimCellCollection.h>
7 #include <EventDataModel/MCParticleCollection.h>
```

EDM related headers

```
14    podio::EventStore store;
15    podio::ROOTReader reader;
16    reader.openFile(string(filename));
17    store.setReader(&reader);
```

Open file in ROOTReader and then connect to EventStore

```
23    for (int jentry=0;jentry<reader.getEntries();++jent
ry)
24    {
25        auto& calohits = store.get<edm::CaloSimCellColle
ction>("calohits");
26        auto& mcparts = store.get<edm::MCParticleCollect
ion>("mcparts");
27
28        float Edep=0;
29        for (int i=0;i<calohits.size();++i)
30        {
31            edm::CaloSimCell calohit = calohits.at(i);
32            float E = calohit.getEdep();
33            short ix = calohit.getIx();
34            short iy = calohit.getIy();
35            short iz = calohit.getIz();
36            Edep += E;
37        }
38        h->Fill(Edep);
39
40        reader.endOfEvent();
41        store.clear();
42    }
```

`reader.getEntries()` to findout the total number of entries

In the current event, fetch the needed collections, e.g. `calohits` for calo simulation info, `mcparts` for mc truth particles

calohits is the collection of all the simulation information for each crystal

Prepare the next event. The next `store.get()` will then read from the next event.

Demo Code: Read with TTreeReader

Limitations:
some complex members may not work.

UseTTreeReader.C

```
#include <iostream>
#include "TCanvas.h"
#include "TH1.h"
#include "TTree.h"
#include "TFile.h"
#include <TTreeReader.h>
#include <TTreeReaderValue.h>
#include <TTreeReaderArray.h>

using namespace std;

void UseTTreeReader(TString filename)
{
   TTreeReader fReader; //the tree reader
   auto* calohits_edep = new TTreeReaderArray<float>(fReader,"calohits.edep");
   auto* calohits_ix = new TTreeReaderArray<short>(fReader,"calohits.ix");
   auto* calohits_iy = new TTreeReaderArray<short>(fReader,"calohits.iy");
   auto* calohits_iz = new TTreeReaderArray<short>(fReader,"calohits.iz");

   TFile* f = new TFile(filename,"read");
   TTree* ReadTree = (TTree*)f->Get("events");
   fReader.SetTree(ReadTree);

   int nentries = fReader.GetEntries(true);

   TH1F* h = new TH1F("h","h",100,0,120);
   for (int jentry=0;jentry<nentries;++jentry)
   {
      fReader.SetEntry(jentry);
      float Edep=0;
      for (int i=0;i<calohits_edep->GetSize();++i)
      {
         Edep += (*calohits_edep)[i];
      }
      h->Fill(Edep);
   }
   TCanvas* c = new TCanvas("c","c",800,600);
   h->Draw();
   gPad->SetLogy();
   c->SaveAs("edep.png");
}
```

can be found at https://code.ihep.ac.cn/herdos/offline/-/tree/34-enrich-examples/Examples/StandAloneAnalysis/

```
6 #include <TTreeReader.h>
7 #include <TTreeReaderValue.h>
8 #include <TTreeReaderArray.h>
```

TTreeReader related headers

```
14    TTreeReader fReader; //the tree reader
15    auto* calohits_edep = new TTreeReaderArray<float>(f
   Reader,"calohits.edep");
16    auto* calohits_ix = new TTreeReaderArray<short>(fRe
   ader,"calohits.ix");
17    auto* calohits_iy = new TTreeReaderArray<short>(fRe
   ader,"calohits.iy");
18    auto* calohits_iz = new TTreeReaderArray<short>(fRe
   ader,"calohits.iz");
```

Define branches to read.
The branch names are in the form of `collection.member`

```
20    TFile* f = new TFile(filename,"read");
21    TTree* ReadTree = (TTree*)f->Get("events");
22    fReader.SetTree(ReadTree);
```

Connect the reader with event tree

```
24   int nentries = fReader.GetEntries(true);
25
26   TH1F* h = new TH1F("h","h",100,0,120);
27   for (int jentry=0;jentry<nentries;++jentry)
28   {
29      fReader.SetEntry(jentry);
30      float Edep=0;
31      for (int i=0;i<calohits_edep->GetSize();++i)
32      {
33         Edep += (*calohits_edep)[i];
34      }
35      h->Fill(Edep);
36   }
```

Total entries

Read the entry

each element in calohits_edep is the energy deposition in a crystal

MC sample in database

- Currently standard proton, electron, gamma sample are registered in database:
  - https://dms.herd.ihep.ac.cn
  - (files stored under Rucio)
- Automated MC sample request and production system under development
  - Currently MC sample request could be send to herd@maillist.ihep.ac.cn

Summary

- The full chain of HERDOS usage is demonstrated including
  - software environment
  - running simulation, digitization & reconstruction
  - reading the output files in analysis
- A more detailed user training is planned and under preparation