

This work was supported by RFBR according to the research project No 18-02-40044

The MpdMiniDst data format (Part 2)

Grigory Nigmatkulov¹ and Pavel Batyuk²

1. National Research Nuclear University MEPhI
2. Joint Institute for Nuclear Research

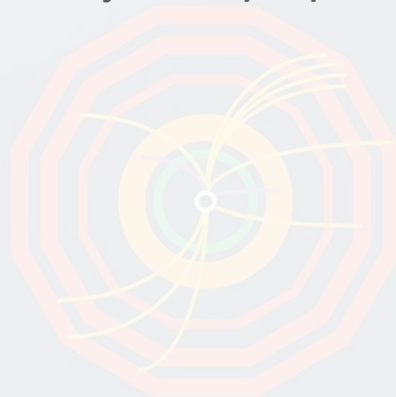
E-mail: nigmatkulov@gmail.com , ganigmatkulov@mephi.ru



August 6, 2020

Outline

- Motivation
- How to compile format and process miniDst on your *laptop*
- Usage example
- Summary and discussions



Motivation

The MpdMiniDst format structure was described in the previous presentation and can be found [here](#). The example of how to run codes in the MpdRoot environment was also presented.

LESSON FOR TODAY

How to write a simple analysis macro and process miniDst files with only ROOT installed on *laptop* (PC, computer farm, etc)

In order to workout next example one needs (prerequisites):

- [ROOT](#) installed
- Makefile
- g++ (≥ 4.8) or clang

How to compile format and process miniDst on your laptop

```
// ROOT headers
#include "Rtypes.h"
#include "TChain.h"
#include "TFile.h"
#include "TVector3.h"
#include "TH1F.h"
// MpdMiniDst headers (with relative path or just header name, in case of compiled macro)
#include "MpdMiniEvent/MpdMiniDstReader.h"
#include "MpdMiniEvent/MpdMiniDst.h"
#include "MpdMiniEvent/MpdMiniEvent.h"
#include "MpdMiniEvent/MpdMiniTrack.h"
#include "MpdMiniEvent/MpdMiniTrackCovMatrix.h"
#include "MpdMiniEvent/MpdMiniBTofHit.h"
#include "MpdMiniEvent/MpdMiniBTofPidTraits.h"
#include "MpdMiniEvent/MpdMiniBECalCluster.h"
#include "MpdMiniEvent/MpdMiniFHCalHit.h"
#include "MpdMiniEvent/MpdMiniMcEvent.h"
#include "MpdMiniEvent/MpdMiniMcTrack.h"
// Load MpdMiniDst library
R_LOAD_LIBRARY(MpdMiniEvent/libMpdMiniDst)
//
void miniExampleVanilla(const Char_t* inFileName = "inputfile.MiniDst.root") {
    MpdMiniDstReader* miniDstReader = new MpdMiniDstReader(inFileName); // Instantiate reader
    miniDstReader->Init(); // Reader initialization
    miniDstReader->SetStatus("...", 0); // One can specify branches to read
    miniDstReader->SetStatus("Event", 1); // Turn ON specific branch
    miniDstReader->SetStatus("Track", 1);
    miniDstReader->SetStatus("BTofPidTraits", 1); // Turn ON specific branch
    miniDstReader->SetStatus("BTofHit", 0); // Turn OFF specific branch
    Long64_t events2read = miniDstReader->chain()->GetEntries(); // Retrieve events in tree
    for (Long64_t i = 0; i < events2read; i++) { // Loop over events
        Bool_t isOk = miniDstReader->readMiniEvent(i); // Read next event
        MpdMiniDst *dst = miniDstReader->miniDst(); // Retrieve current miniDst
        MpdMiniEvent *event = dst->event(); // Get MiniEvent
        Float_t z = event->primaryVertex().Z(); // Get primary vertex z-position
        for (Int_t j = 0; j < dst->numberOfTracks(); j++) { // Track loop
            MpdMiniTrack *miniTrack = dst->track(j); // Retrieve j-th miniTrack
            Float_t gMom = miniTrack->gMom(); // Retrieve global/kalman track 3-momentum
            if (miniTrack->isBTofTrack()) { // Check if track has TOF matching information
                MpdMiniBTofPidTraits *trait = dst->btofPidTraits(miniTrack->bTofPidTraitsIndex()); // TOF-matching info
                Float_t beta = trait->beta(); // Retrieve beta associated with the track
            }
        }
    }
    miniDstReader->Finish(); // Finalize miniDst reader (remove pointers, etc)
}
```

Example structure:

dir/

miniExampleVanilla.C

MpdMiniEvent/

MpdMiniDst.h(cxx)

MpdMiniEvent.h(cxx)

MpdMiniTrack.h(cxx)

...

Makefile

How to compile format and process miniDst on your laptop

- First of all one need to compile the miniDst codes in **MpdMiniEvent/** directory:

```
[term MpdMiniEvent/] make
```

- Then from the **dir/** directory one can simply execute a macro using:

```
[term dir/] root  
[0] .L miniExampleVanilla.C  
[1] miniExampleVanilla("/path/to/file.MiniDst.root")
```

Or

```
[term dir/] root 'miniExampleVanilla.C("/path/to/file.MiniDst.root")'
```

- The other option to provide a list of MiniDst files as an input with **.lis** or **.list** extension

```
[term dir/] root 'miniExampleVanilla.C("inputfile.list")'
```

Keep .MiniDst.root extension in the filename is important

Summary

1. We presented how to analyze miniDst in a standalone mode
2. Feedback is appreciated

Discussions

1. Fit tracks to the primary vertex
 - a. Tracking expert (Alexandr Zinchenko) refused to do it by himself but promised to provide codes
 - b. Waiting for the codes from Alexandr
2. Store MC tracks that are the decay products of long-lived particles
 - a. What is the MotherID() key for such tracks?
3. How the tracking is performed? Can expert give a presentation about it?
4. How the TOF-matching is performed? Can expert give a presentation about it?
5. What triggers are planned? What information will be stored?