

# Online data processing and monitoring of the BM@N experiment

Ilnur Gabdrakhmanov  
in collaboration with  
Sergei Merts, Andrey Driuk, Konstantin Mashitsin

Joint Institute for Nuclear Research, Laboratory of High Energy Physics

LXXII International conference "Nucleus-2022: Fundamental problems and applications"  
Moscow  
July 15, 2022



Ilnur  
Gabdrakhmanov  
in collaboration  
with  
Sergei Merts,  
Andrey Driuk,  
Konstantin  
Mashitsin

Introduction

Codebase

Preliminary

Monitoring  
workflow

Decoding

Hardcoded  
histograms

External tools

Representation  
examples (Basic)

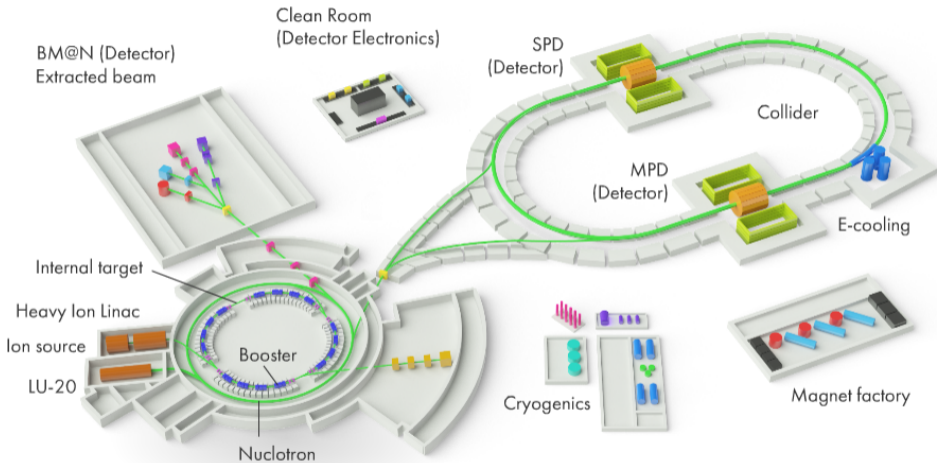
General QA

Custom histograms

Examples

Conclusion

# Nuclotron based Ion Collider fAcility complex



Online data processing and monitoring of the BM@N experiment

**Ilmur Gabdrakhmanov**  
in collaboration with  
**Sergei Merts, Andrey Driuk, Konstantin Mashitsin**

## Introduction

Codebase  
Preliminary

## Monitoring workflow

Decoding

## Hardcoded histograms

## External tools

## Representation examples (Basic)

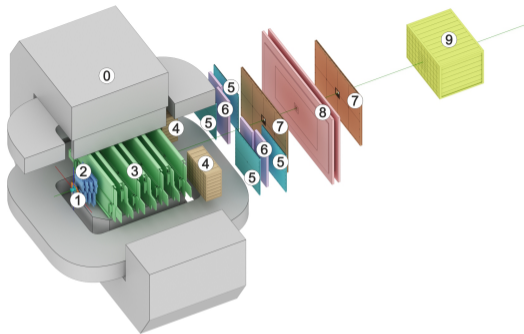
## General QA

Custom histograms  
Examples

## Conclusion



# Baryonic Matter at Nuclotron



- Magnet SP-41 (0)
- Triggers: BD + SiD (1)
- Forward Silicon (2)
- GEM (3)
- ECAL (4)
- CSC 1x1 m<sup>2</sup> (5)
- TOF 400 (6)
- CSC 2x1.5 m<sup>2</sup> (7)
- TOF 700 (8)
- ZDC (9)

Online data processing and monitoring of the BM@N experiment

Ilnur Gabdrakhmanov in collaboration with Sergei Merts, Andrey Driuk, Konstantin Mashitsin

## Introduction

Codebase  
Preliminary

## Monitoring workflow

Decoding

## Hardcoded histograms

## External tools

## Representation examples (Basic)

## General QA

Custom histograms  
Examples

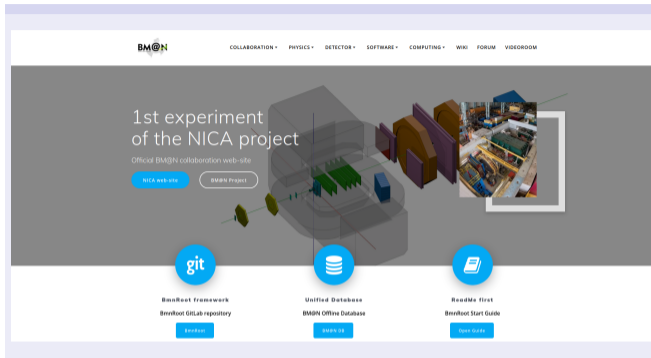
## Conclusion



## Subsystems

- ▶ Trigger system:
  - ▶ Beam counters
  - ▶ Barrel detector
- ▶ Tracking system:
  - ▶ Forward Silicon
  - ▶ GEM (Gas Electron Multipliers)
  - ▶ CSC (Cathode Strip Chambers)
- ▶ Identification(time-of-flight) system:
  - ▶ ToF-400
  - ▶ ToF-700
- ▶ Calorimeters:
  - ▶ ZDC (Zero Degree Calorimeter)
  - ▶ ECAL (Electromagnetic Calorimeter)

# BM@N Framework BMNROOT



BM@N experiment home web-page:

<https://bmn.jinr.ru>

- ▶ News
- ▶ Software repositories
- ▶ Software tests
- ▶ Forums
- ▶ Database for physics run
- ▶ E.t.c.

## Benefits:

- ▶ Inherits basic properties from FairRoot (<https://fairroot.gsi.de/>), C++ classes
- ▶ Detector composition and geometry; particle propagation by GEANT3/4
- ▶ Advanced detector response functions, realistic tracking and PID included
- ▶ Event display for Monte-Carlo and experimental data
- ▶ QA system

BmnROOT repository

<https://git.jinr.ru/nica/bmnroot>

Online data processing and monitoring of the BM@N experiment

Ilnur Gabdrakhmanov in collaboration with Sergei Merts, Andrey Driuk, Konstantin Mashitsin

Introduction

Codebase

Preliminary

Monitoring workflow

Decoding

Hardcoded histograms

External tools

Representation examples (Basic)

General QA

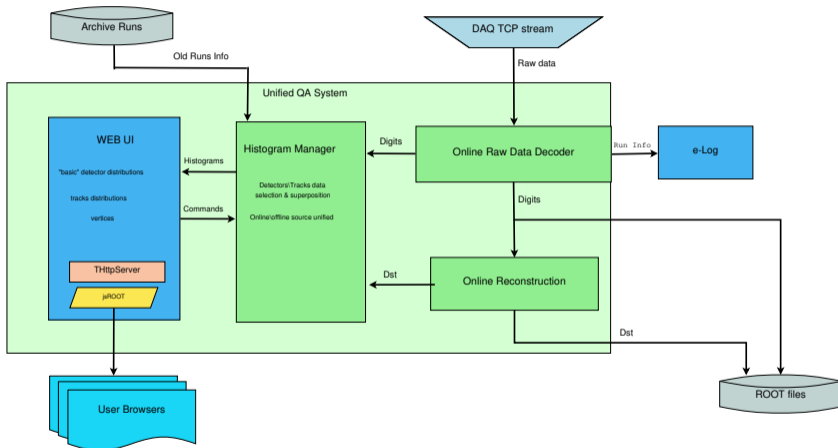
Custom histograms

Examples

Conclusion



# General system scheme



Online data processing and monitoring of the BM@N experiment

Ilnur Gabdrakhmanov  
in collaboration with  
Sergei Merts,  
Andrey Driuk,  
Konstantin Mashitsin

## Introduction

Codebase  
Preliminary

## Monitoring workflow

Decoding

## Hardcoded histograms

## External tools

## Representation examples (Basic)

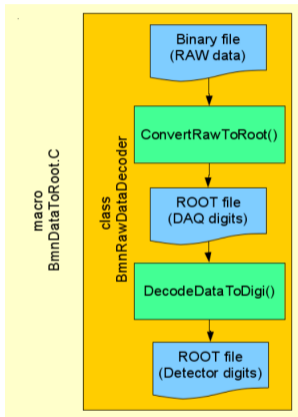
## General QA

Custom histograms  
Examples

## Conclusion



# Decoding scheme



## First step (Data Converter):

- ▶ Read a **binary data file** with RAW-data.
- ▶ Parse the data blocks: **run/spill/event/module**.
- ▶ Create «**DAQ-digits**» (ADC, TDC, TQDC, HRB, SYNC, etc.) accordingly **DAQ-data-format** and write them into a tree.

## Second step (Data Decoder):

- ▶ Read **detector mappings** (channel-to-strip) from the **Unified Database**
- ▶ Calculate **pedestals** and **common modes** of channels
- ▶ Clear **noisy** channels
- ▶ Decode **DAQ-digits** into **detector-digits** (BmnGemDigit, BmnTofDigit, etc.)
- ▶ Write the tree with **detector-digits** to a ROOT-file

# Basic QA frontend with hardcoded histograms

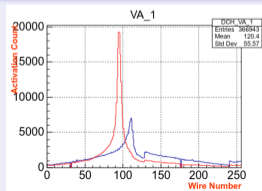
## Implementation details:

- ◇ The data processed and transferred from the previous stage is used to fill ROOT histograms. Which in turn are sent to the end users via http.
- ◇ CERN jsROOT library is used to transform the ROOT object to the html histograms.
- ◇ Base class for histogram sets BmnHist is used in:
  - ▷ BmnHistTrigger
  - ▷ BmnHistGem
  - ▷ BmnHistToF
  - ... ..

Thus addition of the new detector histogram set is rather simple.

## Reference run:

- ✓ Ref run imposition implemented
- ✓ Autoselection of similar runs is implemented



Online data  
processing and  
monitoring of the  
BM@N experiment

Ilnur  
Gabbrakhmanov  
in collaboration  
with  
Sergei Merts,  
Andrey Driuk,  
Konstantin  
Mashitsin

Introduction

Codebase

Preliminary

Monitoring  
workflow

Decoding

Hardcoded  
histograms

External tools

Representation  
examples (Basic)

General QA

Custom histograms

Examples

Conclusion



# Fine grain selection (station/plane/strip):

Plane Index

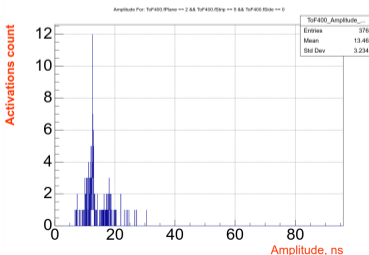
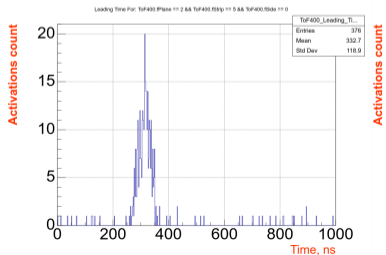
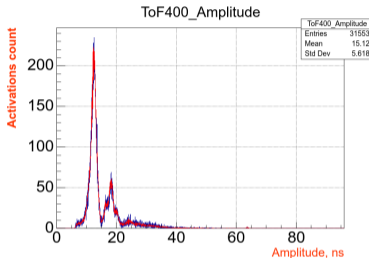
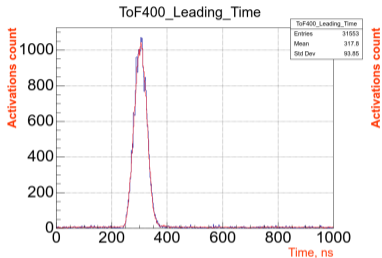
Strip Index

Side Index

Run Index 1480

**Change Selection (-1 => All)**

**Select Reference Run**



Online data processing and monitoring of the BM@N experiment

Ilnur Gabdrakhmanov in collaboration with Sergei Merts, Andrey Driuk, Konstantin Mashitsin

Introduction

Codebase

Preliminary

Monitoring workflow

Decoding

Hardcoded histograms

External tools

Representation examples (Basic)

General QA

Custom histograms

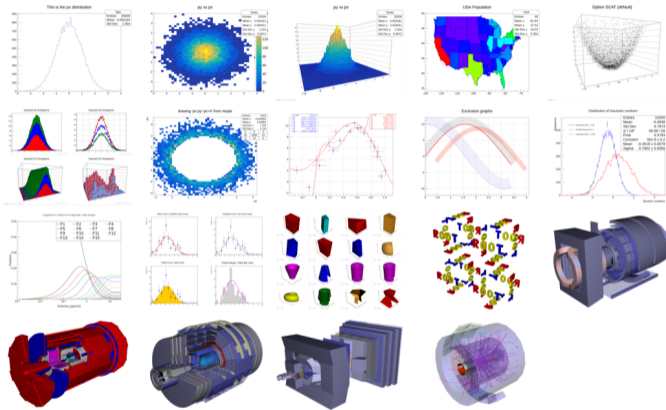
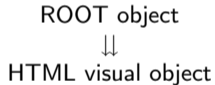
Examples

Conclusion





CERN jsROOT library:



jsROOT website  
<https://root.cern.ch/js/>

Online data processing and monitoring of the BM@N experiment

**Ilnur Gabdrakhmanov**  
 in collaboration with  
**Sergei Merts, Andrey Driuk, Konstantin Mashitsin**

Introduction

Codebase

Preliminary

Monitoring workflow

Decoding

Hardcoded histograms

External tools

Representation examples (Basic)

General QA

Custom histograms

Examples

Conclusion



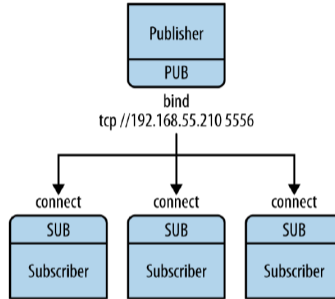


Lightweight high-speed library for network messaging

- Variety of transports: TCP, interprocess, inproc
- Automatic queue and buffer managing
- Many usable messaging patterns

ZeroMQ website

<https://zeromq.org>



Online data processing and monitoring of the BM@N experiment

Ilnur Gabdrakhmanov in collaboration with Sergei Merts, Andrey Driuk, Konstantin Mashitsin

Introduction

Codebase

Preliminary

Monitoring workflow

Decoding

Hardcoded histograms

External tools

Representation examples (Basic)

General QA

Custom histograms

Examples

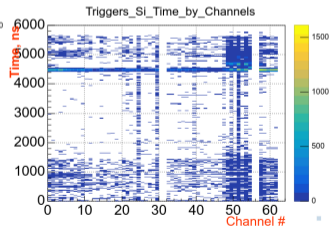
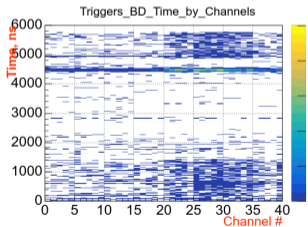
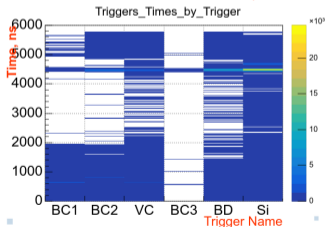
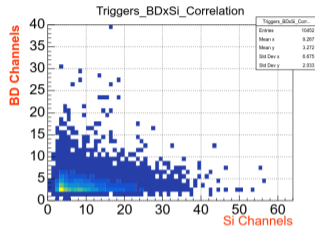
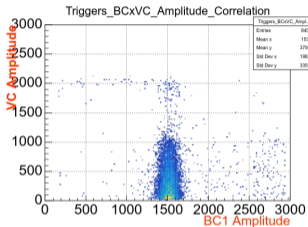
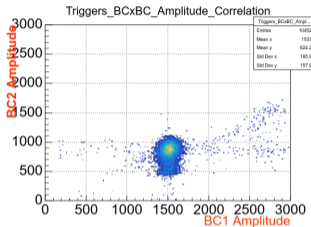
Conclusion



# Trigger distributions during the 2018 technical run:

Reset

Select Reference Run Run 3946, beam Ar, energy 5.2, target C, Voltage 77.597222



Online data processing and monitoring of the BM@N experiment

Ilnur Gabdrakhmanov  
in collaboration with  
Sergei Merts,  
Andrey Driuk,  
Konstantin Mashitsin

Introduction

Codebase

Preliminary

Monitoring workflow

Decoding

Hardcoded histograms

External tools

Representation examples (Basic)

General QA

Custom histograms

Examples

Conclusion



## Existing online processing frameworks

### ■ TDAQ (ATLAS)

- tightly integrated with other ATLAS software
- thus it is rather difficult to deploy in other program environment

### ■ FairMQ (GSI FAIR)

- seems to be quite flexible in deployment and settings
- but requires rewriting a lot of existing bmnroot analysis code (based on FairTask's)

Online data  
processing and  
monitoring of the  
BM@N experiment

Ilnur  
Gabbrakhmanov  
in collaboration  
with  
Sergei Merts,  
Andrey Driuk,  
Konstantin  
Mashitsin

Introduction

Codebase

Preliminary

Monitoring  
workflow

Decoding

Hardcoded  
histograms

External tools

Representation  
examples (Basic)

General QA

Custom histograms

Examples

Conclusion



## FairRoot way of analysis via FairTask's (Extensively being used in the BmnRoot)

- FairRunAna - task manager class
- FairSource - abstract class for a data source
- FairSink - abstract class for a data destination manager

### Typical analysis macro workflow:

- ▷ BmnFileSource/FairFileSource (input data file )
- ▷ Task1 (executed event-by-event)
- ▷ Task2
- ▷ Task3
- ▷ ...
- ▷ FairRootFileSink (output data file)

#### Introduction

Codebase

Preliminary

#### Monitoring workflow

Decoding

#### Hardcoded histograms

#### External tools

#### Representation examples (Basic)

#### General QA

Custom histograms

Examples

#### Conclusion



## Simplest way to move existing reconstruction code to online

### ZMQ transfer classes for FairRunAna

- BmnMQSource - ZeroMQ SUB socket<sup>1</sup> based source class
- BmnMQSink - ZeroMQ PUB socket based sink class

### Benefits

- No need to rewrite existing bmnroot analysis code
- It became possible to combine several analysis macros by source/sink network interfaces

---

<sup>1</sup><https://zeromq.org>

# BmnRoot QA structure

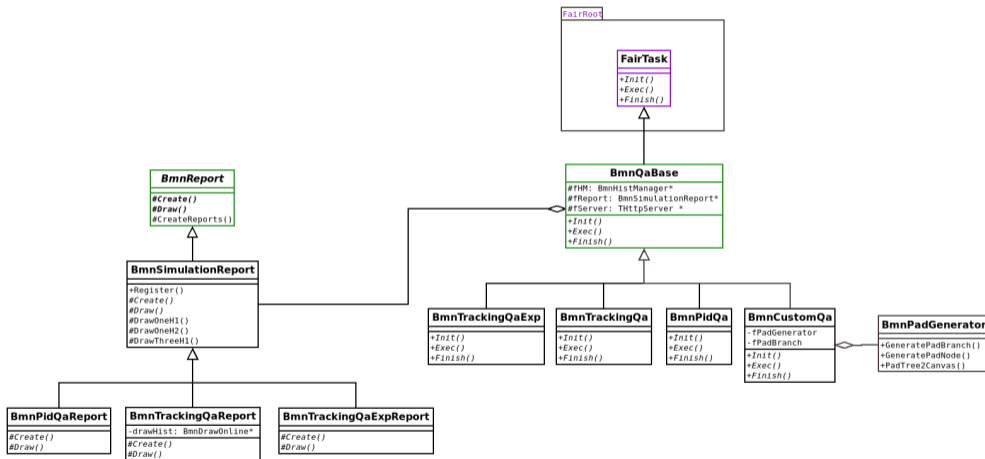


Figure: QA main classes (green ones were forked from CbmRoot)

Online data processing and monitoring of the BM@N experiment

Ilnur Gabdrakhmanov in collaboration with Sergei Merts, Andrey Driuk, Konstantin Mashitsin

Introduction

Codebase

Preliminary

Monitoring workflow

Decoding

Hardcoded histograms

External tools

Representation examples (Basic)

General QA

Custom histograms

Examples

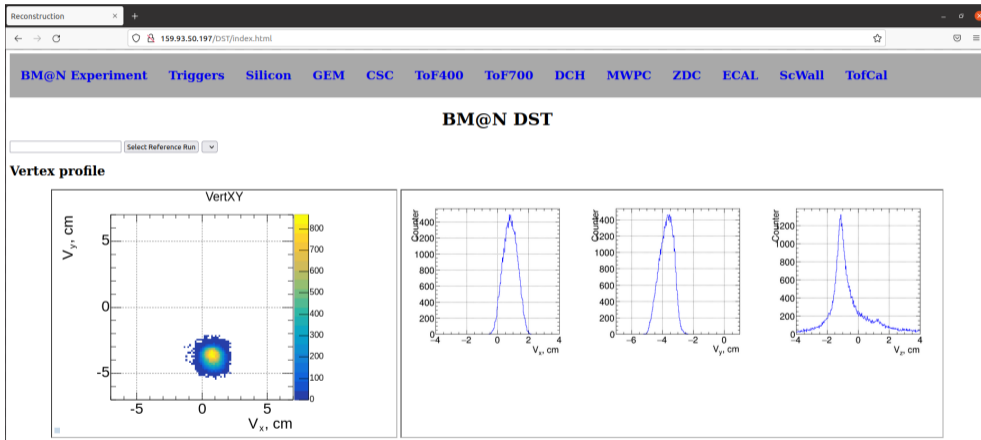
Conclusion



# Live example of the online reconstruction page

Online data processing and monitoring of the BM@N experiment

Ilnur Gabdrakhmanov  
in collaboration with  
Sergei Merts,  
Andrey Driuk,  
Konstantin Mashitsin



## Introduction

Codebase

Preliminary

## Monitoring workflow

Decoding

## Hardcoded histograms

## External tools

## Representation examples (Basic)

## General QA

Custom histograms

Examples

## Conclusion





# Custom «no code» histograms. Motivation

## Why?

Experiment upgrade as well as conduction of two experimental setups require distribution of work on the development of the online QA system.

Namely each detector team should be able to extend system's functionality easily.

Online data  
processing and  
monitoring of the  
BM@N experiment

**Ilnur  
Gabbrakhmanov**  
in collaboration  
with  
**Sergei Merts,  
Andrey Driuk,  
Konstantin  
Mashitsin**

### Introduction

Codebase

Preliminary

### Monitoring workflow

Decoding

### Hardcoded histograms

### External tools

### Representation examples (Basic)

### General QA

**Custom histograms**

Examples

### Conclusion



### Why?

Experiment upgrade as well as conduction of two experimental setups require distribution of work on the development of the online QA system.

Namely each detector team should be able to extend system's functionality easily.

### Main objectives:

- Move monitoring configuration outside of the code
- Make addition of histogram simple and flexible (It should not require code rebuild)
- Implement filling logic configurable as well (thanks to ROOT TTree::Draw text parser it was possible)

### Why?

Experiment upgrade as well as conduction of two experimental setups require distribution of work on the development of the online QA system.

Namely each detector team should be able to extend system's functionality easily.

### Main objectives:

- Move monitoring configuration outside of the code
- Make addition of histogram simple and flexible (It should not require code rebuild)
- Implement filling logic configurable as well (thanks to ROOT TTree::Draw text parser it was possible)

### Implementation

BmnPadGenerator class - creates a pad structure in the canvas on the basis of json scheme.

Test code example:

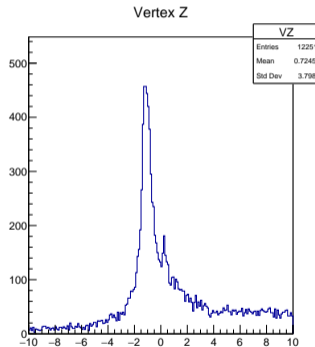
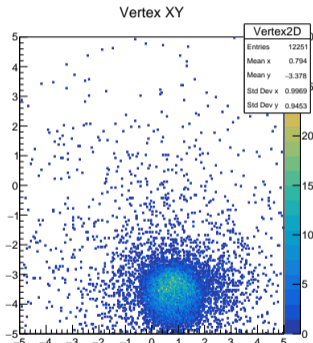
```
BmnPadGenerator *g = new BmnPadGenerator();  
g->LoadPTFrom(FileName);  
BmnPadBranch * br = g->GetPadBranch();  
TCanvas* can = new TCanvas("canHits", "", 1920, 1080);  
g->PadTree2Canvas(br, can);  
BmnHist::DrawPadTree(br);
```

# Simple configuration

## JSON scheme:

```
{
  "Name": "Custom canvas",
  "Title": "Custom Canvas",
  "DivX": "2",
  "DivY": "1",
  "Pads": [
    {
      "Class": "TH2F",
      "Name": "Vertex2D",
      "Title": "Vertex XY",
      "Variable": "BmVertex.fY:BmVertex.fX",
      "Selection": "(BmVertex.fZ>-10 && BmVertex.fZ<10)",
      "Options": "colz",
      "Dimensions": [
        200,
        -5,
        5,
        200,
        -5,
        5
      ]
    },
    {
      "Class": "TH1F",
      "Name": "VZ",
      "Title": "Vertex Z",
      "Variable": "BmVertex.fZ",
      "Selection": "(BmVertex.fZ>-10 && BmVertex.fZ<10)",
      "Dimensions": [
        200,
        -10,
        10
      ]
    }
  ]
}
```

## Canvas structure:



Online data processing and monitoring of the BM@N experiment

Ilnur Gabdrakhmanov  
in collaboration with  
Sergei Merts,  
Andrey Driuk,  
Konstantin Mashitsin

Introduction

Codebase

Preliminary

Monitoring workflow

Decoding

Hardcoded histograms

External tools

Representation examples (Basic)

General QA

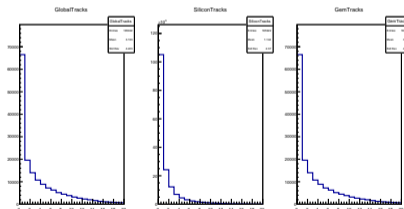
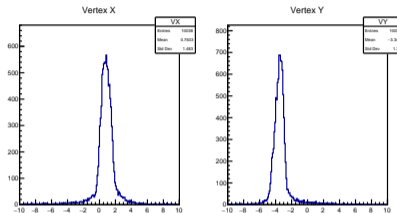
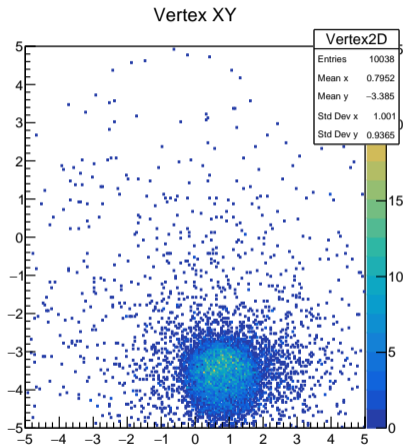
Custom histograms

Examples

Conclusion



# More complex configuration



Online data processing and monitoring of the BM@N experiment

Ilnur Gabdrakhmanov  
in collaboration with  
Sergei Merts,  
Andrey Driuk,  
Konstantin Mashitsin

Introduction

Codebase

Preliminary

Monitoring workflow

Decoding

Hardcoded histograms

External tools

Representation examples (Basic)

General QA

Custom histograms

Examples

Conclusion



# Conclusion

- ◇ Unified online/offline QA system is being developed in the framework of the bmnroot package
- ◇ ZeroMQ network transfer classes were developed for FairRunManager based analysis
- ◇ "No code" approach were developed in order to simplify extension of the system

Online data  
processing and  
monitoring of the  
BM@N experiment

**Ilnur  
Gabbrakhmanov**  
in collaboration  
with  
**Sergei Merts,  
Andrey Driuk,  
Konstantin  
Mashitsin**

Introduction

Codebase

Preliminary

Monitoring  
workflow

Decoding

Hardcoded  
histograms

External tools

Representation  
examples (Basic)

General QA

Custom histograms

Examples

Conclusion



## Conclusion

- ◇ Unified online/offline QA system is being developed in the framework of the bmnroot package
- ◇ ZeroMQ network transfer classes were developed for FairRunManager based analysis
- ◇ "No code" approach were developed in order to simplify extension of the system

Thanks for your attention!

Online data  
processing and  
monitoring of the  
BM@N experiment

**Ilnur  
Gabbrakhmanov**  
in collaboration  
with  
**Sergei Merts,  
Andrey Driuk,  
Konstantin  
Mashitsin**

Introduction

Codebase

Preliminary

Monitoring  
workflow

Decoding

Hardcoded  
histograms

External tools

Representation  
examples (Basic)

General QA

Custom histograms

Examples

Conclusion

