

Online monitoring in Xe run

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

Joint Institute for Nuclear Research, Laboratory of High Energy Physics

IX Collaboration Meeting of the BM@N Experiment at the NICA Facility
Dubna
September 16, 2022



Online monitoring
in Xe run

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

Introduction

Codebase
Preliminary

Monitoring
workflow

Decoding

Hardcoded
histograms

External tools

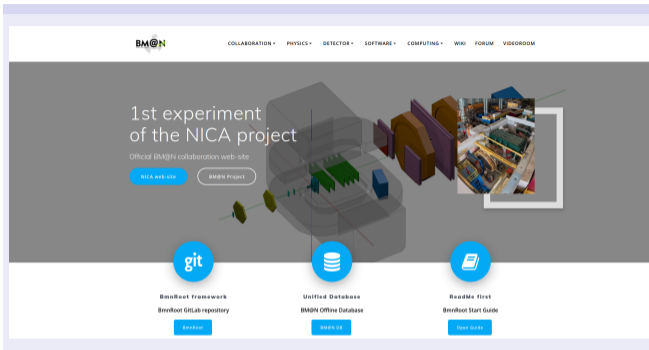
Representation
examples (Basic)

General QA

Custom histograms
Examples

Conclusion

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 monitoring
in Xe run

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

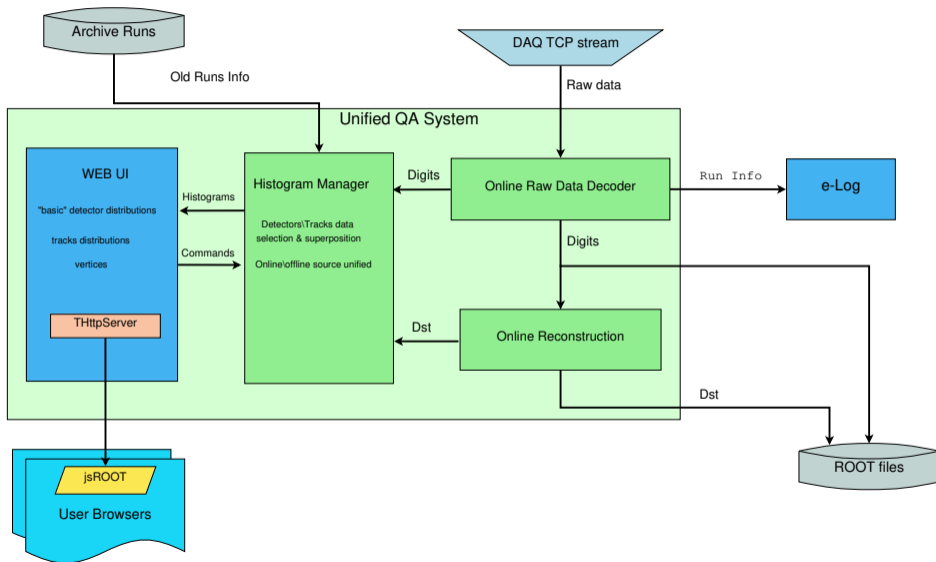
Custom histograms

Examples

Conclusion



General system scheme



Online monitoring
in Xe run

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

Introduction

Codebase
Preliminary

Monitoring
workflow

Decoding

Hardcoded
histograms

External tools

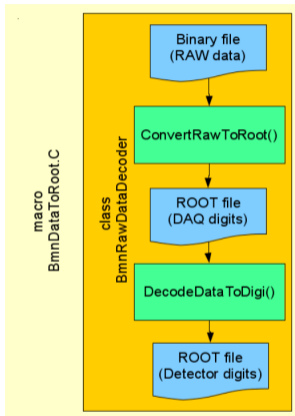
Representation
examples (Basic)

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

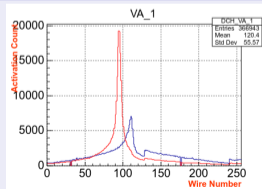
Implementation details:

- ◇ 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 monitoring
in Xe run

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

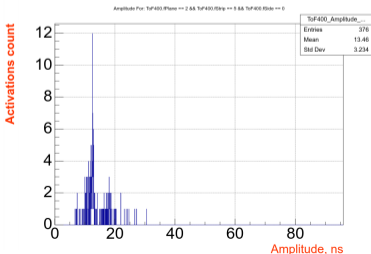
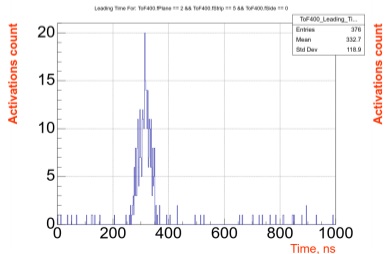
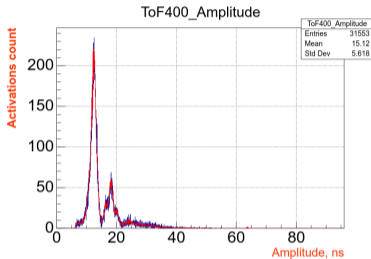
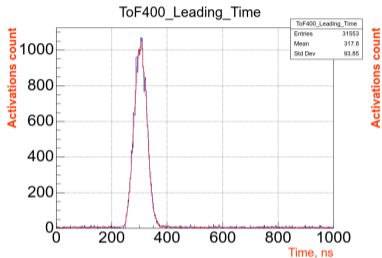
Examples

Conclusion



Fine grain selection (station/plane/strip):

Plane Index
Strip Index
Side Index
Run Index



Online monitoring
in Xe run

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

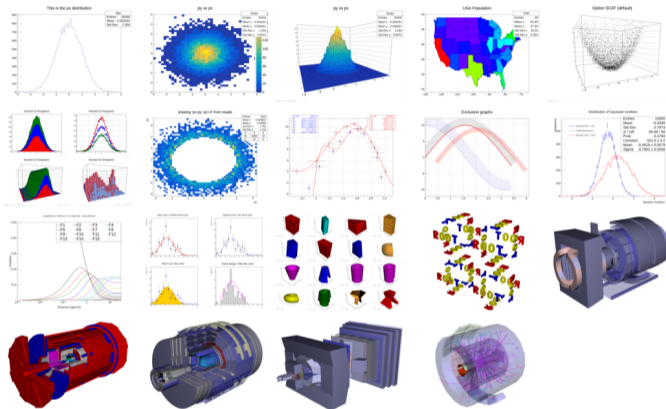


CERN jsROOT library:

ROOT object



HTML visual object



jsROOT website

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

Online monitoring
in Xe run

Ilnur
Gabrakhmanov
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

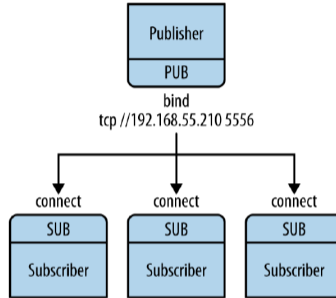


Lightweight high-speed library for network messaging

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

[ZeroMQ website](https://zeromq.org)

<https://zeromq.org>



Online monitoring
in Xe run

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

Introduction

Codebase
Preliminary

Monitoring
workflow

Decoding

Hardcoded
histograms

External tools

Representation
examples (Basic)

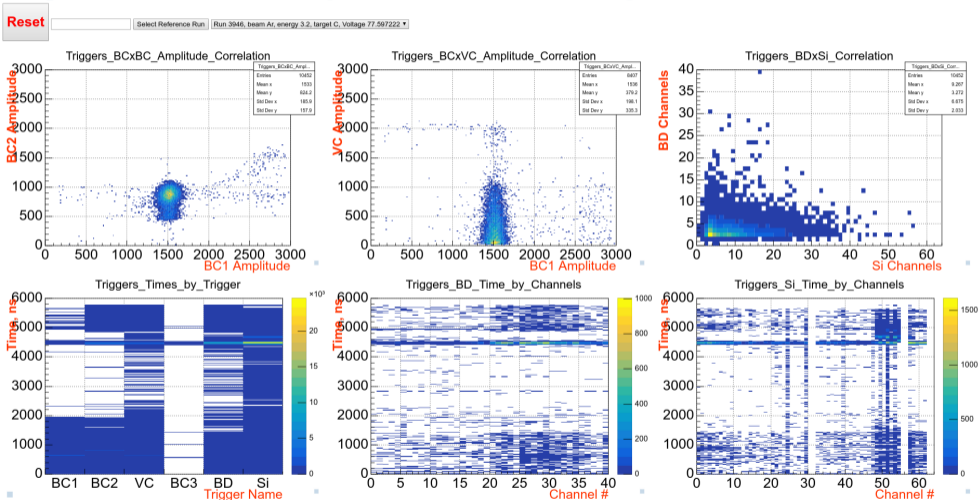
General QA

Custom histograms
Examples

Conclusion



Trigger distributions during the 2018 technical run:



Online monitoring
in Xe run

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)

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

Less code \rightarrow Less errors

ZMQ transfer classes for FairRunAna

- BmnMQSource - ZeroMQ SUB socket¹ based source class
- BmnMQSink - ZeroMQ PUB socket based sink class

Benefits

- No need to rewrite existing bmnroot analysis code. (No need to touch any working task)
- It became possible to combine several analysis macros by source/sink network interfaces

Online monitoring
in Xe run

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

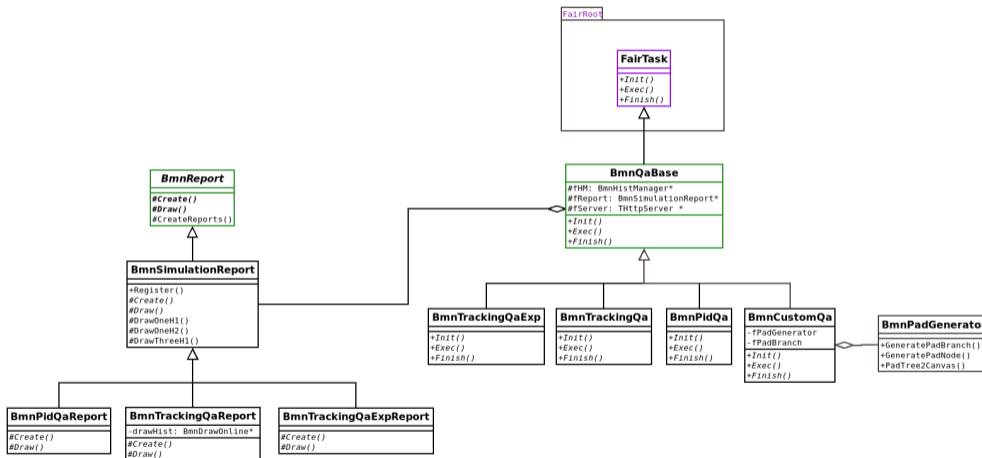


¹<https://zeromq.org>

BmnRoot QA structure

Online monitoring
in Xe run

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

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



Decoding & Monitoring status

| Detector | Decoding | Reco/HitMaker | Monitoring: Digi | DST |
|-----------|-------------|-----------------|------------------|-------------|
| Triggers | ok (no map) | - | ok | - |
| Silicon | ok (no map) | ok | ok | ok |
| SiBT | in progress | ok | ok | ok |
| GEM | ok | ok | ok | ok |
| CSC | ok (no map) | ok | ok | ok |
| DCH | ok (no map) | ok | ok | in progress |
| MWPC | ok (no map) | ok | ok | in progress |
| ToF400 | ok (no map) | ok | ok | ok |
| ToF700 | ok (no map) | ok | ok | ok |
| ScWall | ok (no map) | not implemented | in progress | in progress |
| FHCaI | ok (no map) | ok | in progress | in progress |
| Hodoscope | ok (no map) | not implemented | in progress | in progress |

Profilometers monitoring implemented as a standalone executable. Currently is being rewritten to use general bmnroot ADC processing pipeline and histograms, although the source is separate from the DAQ.

Online monitoring
in Xe run

**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

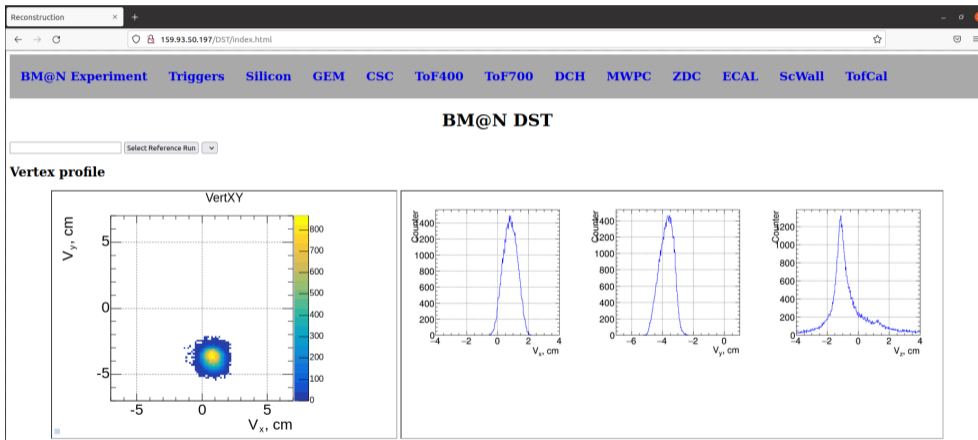
Examples

Conclusion

Live example of the online reconstruction page

Online monitoring
in Xe run

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



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 monitoring
in Xe run

**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



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.

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)

Online monitoring
in Xe run

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



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:

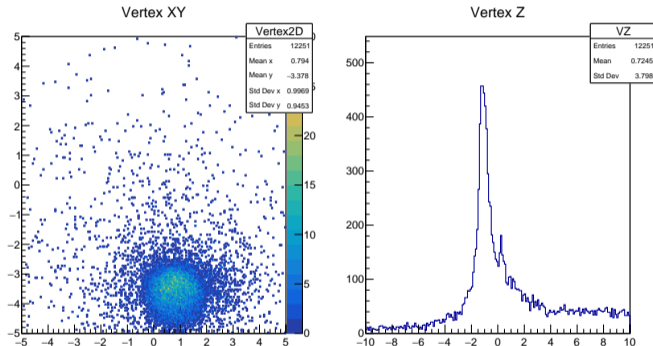
```
BmnPadGenerator *g = new BmnPadGenerator();
g->LoadPTFrom(fileName);
BmnPadBranch * br = g->GetPadBranch();
TCanvas* can = new TCanvas("canHits", "", 1920, 1080);
g->PadTree2Canvas(br, can);
BmnHist::DrawPadTree(br);
```

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:



- ◇ Works well for data in TClonesArray branches
- ◇ Doesn't work for single object branches out of the box (only with additional code for each class)

Online monitoring
in Xe run

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

Introduction

Codebase
Preliminary

Monitoring
workflow

Decoding

Hardcoded
histograms

External tools

Representation
examples (Basic)

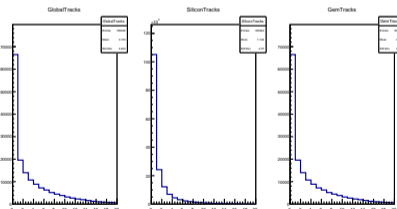
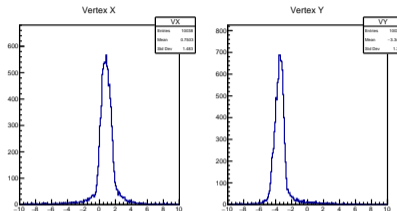
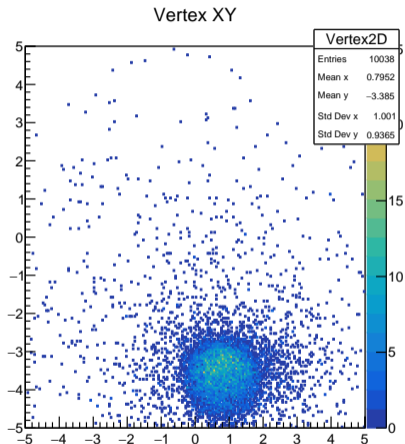
General QA

Custom histograms
Examples

Conclusion



More complex configuration



Online monitoring
in Xe run

**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

- ◇ Purely online QA system based on BmnHist class
 - ▶ currently includes only digits ([1st variant of monitoring](#))
 - ▶ not flexible enough
- ◇ Offline QA system based on BmnQaBase is being unified for online/offline usage:
 - ▶ hardcoded variant currently includes only DST data ([2nd variant](#))
 - ▶ "No code" approach were developed in order to simplify extension of the system([3rd variant](#))
- ◇ ZeroMQ transfer source/sink classes were developed for FairRunManager based analysis.

Online monitoring
in Xe run

**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

- ◇ Purely online QA system based on BmnHist class
 - ▶ currently includes only digits ([1st variant of monitoring](#))
 - ▶ not flexible enough
- ◇ Offline QA system based on BmnQaBase is being unified for online/offline usage:
 - ▶ hardcoded variant currently includes only DST data ([2nd variant](#))
 - ▶ "No code" approach were developed in order to simplify extension of the system([3rd variant](#))
- ◇ ZeroMQ transfer source/sink classes were developed for FairRunManager based analysis.

We would like to encourage all detector groups to send the lists of desired histograms!

Online monitoring
in Xe run

**Ilmur
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

- ◇ Purely online QA system based on BmnHist class
 - ▶ currently includes only digits ([1st variant of monitoring](#))
 - ▶ not flexible enough
- ◇ Offline QA system based on BmnQaBase is being unified for online/offline usage:
 - ▶ hardcoded variant currently includes only DST data ([2nd variant](#))
 - ▶ "No code" approach were developed in order to simplify extension of the system([3rd variant](#))
- ◇ ZeroMQ transfer source/sink classes were developed for FairRunManager based analysis.

We would like to encourage all detector groups to send the lists of desired histograms!

Thanks for your attention!

Online monitoring
in Xe run

**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

