

# DB READER: C++ LIBRARY FOR DATA RETRIEVAL FROM DOSSIER DB

---

Dmitri Konstantinov (CERN, IHEP)

# Motivation

- Several examples when physics tests have a comparison with experimental data:
  - with unclear origin (some hints in file names, text files);
  - from some old physics journals not published in electronic format;
  - some physics paper published in Russian language only.

# Idea

To ensure somehow everyone that in our physics test we use validated data with known origin.

# Possible solutions

- To keep clear formatted README file describing experimental data origin (too simple?)
- To use DOSSIER db and access datasets directly from C++ test code using dedicated library:

Q: How do we do that?

A: Using some connection between dataset used in test with dataset kept in Dossier we retrieve validated dataset from Dossier and fill C++ data class.

# Dataset identifier

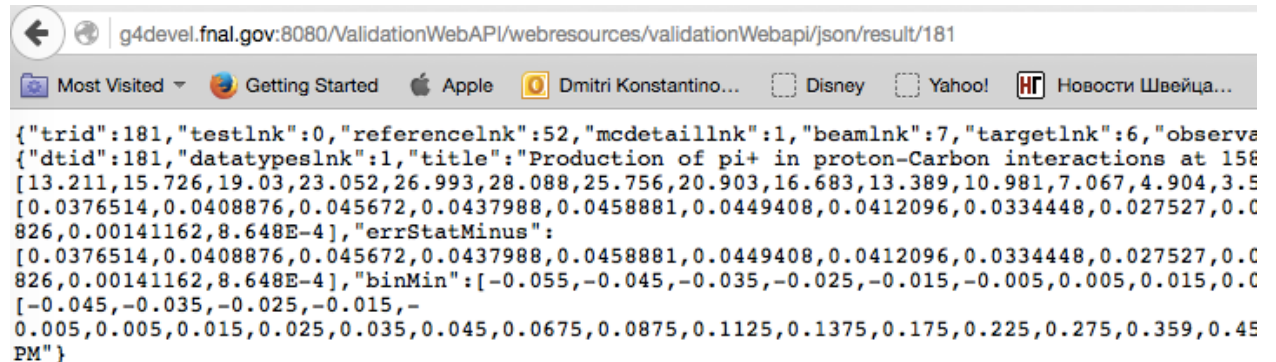
- Id of database table containing META information for given dataset:
  - not elegant solution
  - unique id
  - Implies new requirement for DOSSIER:
    - ***We should never change this ID once Dossier in production.***
- Another solution is to use SQL query based on combination of META information:
  - has to be provided by Dossier Web Application (not implemented):
  - which is supposed to be UNIQUE in this case (1 query = 1 result)
  - not clear how to achieve this.

First solution is chosen.

# DbReader: C++ library

simple C++ library allowing data retrieval using dataset id is written. Data can be retrieved by different methods:

- 1) by http(s) protocol reading JSON file streamed by Dossier web application:



```

{"trid":181,"testlnk":0,"referencelnk":52,"mcdetaillnk":1,"beamlnk":7,"targetlnk":6,"observe
{"dtid":181,"datatypeslnk":1,"title":"Production of pi+ in proton-Carbon interactions at 15E
[13.211,15.726,19.03,23.052,26.993,28.088,25.756,20.903,16.683,13.389,10.981,7.067,4.904,3.5
[0.0376514,0.0408876,0.045672,0.0437988,0.0458881,0.0449408,0.0412096,0.0334448,0.027527,0.C
826,0.00141162,8.648E-4],"errStatMinus":
[0.0376514,0.0408876,0.045672,0.0437988,0.0458881,0.0449408,0.0412096,0.0334448,0.027527,0.C
826,0.00141162,8.648E-4],"binMin":[-0.055,-0.045,-0.035,-0.025,-0.015,-0.005,0.005,0.015,0.C
[-0.045,-0.035,-0.025,-0.015,-
0.005,0.005,0.015,0.025,0.035,0.045,0.0675,0.0875,0.1125,0.1375,0.175,0.225,0.275,0.359,0.45
PM"}

```

dataset is retrieved, parsed and returned as ROOT TGraph or TH object filled with this data

Two external libraries are used:

libCurl - the multiprotocol file transfer library

libjson-cpp - C++ library for interacting with JSON.

# C++ library to retrieve data from DB

Second method: directly accessing DB tables using C API library for PostgreSQL database (part of standard PostgreSQL):

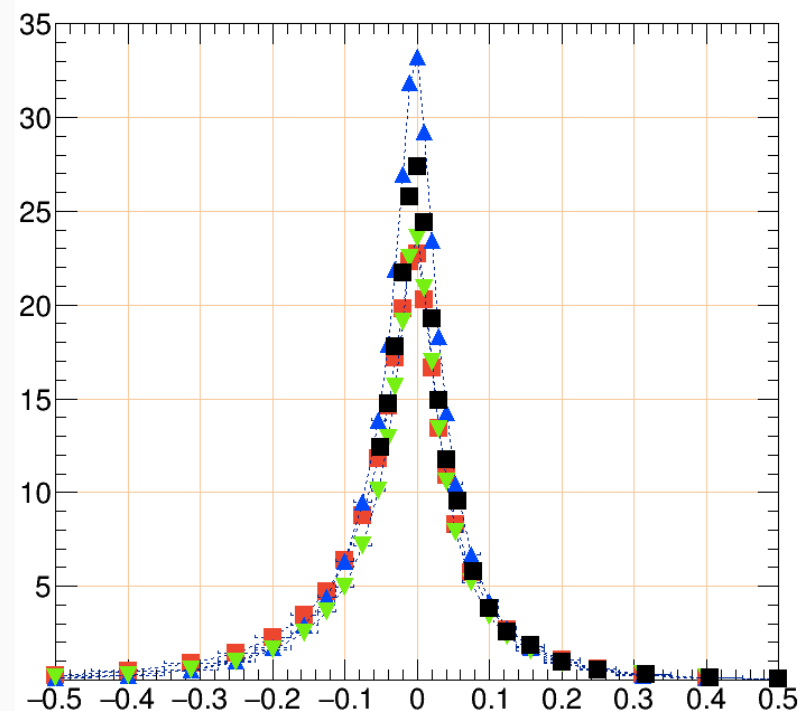
- From the security perspective, this is clearly not ideal, as a typical application was never designed to be utilised this way.

***Implemented but will be removed.***

# DbReader: How to use

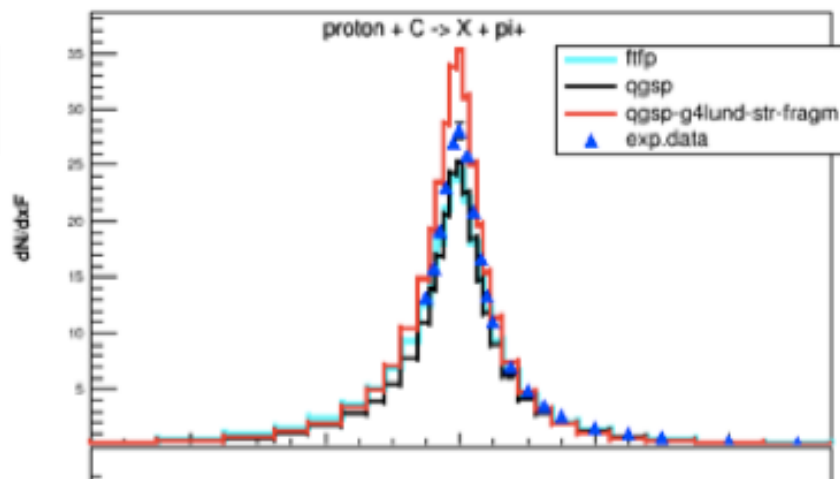
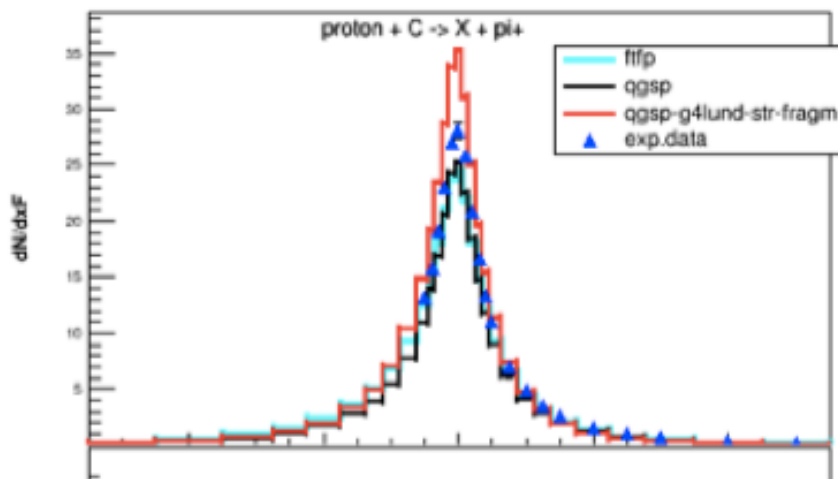
- The library can be used from ROOT 6 macros or your C++ code can be linked with this library.

```
DbReader reader;  
TGraph* data = new TGraphAsymmErrors( reader.getByHTTP(183));  
data->SetMarkerSize(2);  
data->SetMarkerStyle(21);  
data->SetMarkerColor(kBlack);  
data->Draw("P");
```



# DbReader: Testing

DbReader prototype is tested with Geant4 “test19” plotting macro.  
Replacing reading experimental data from file by reading from DOSSIER.



It is also tested by Julia



# Conclusion

- C++ prototype for DB interface is ready and tested.
- We have proved that it can be easily used in physics tests.