Status of Data Preservation

Elena Perez del Rio
For the Offline Group

KLOE-2 General Meeting 21-24 September 2018

Introduction – First steps

- Meeting with data preservation experts to present the use case of KLOE-2 for data preservation (February 2018)
  - Sunje Dallmeier-Tiessen

- Present analysis on datarec/dst → change the output to ROOT-tuple following the structure already present for the HBOOK n-tuple

- Experts from ROOT aiding in the process:
  - Axel Naumman, Olivier Couet and other ROOT developers
  - Meeting in the ROOT User's Workshop September with several experts to discuss our advances and strategies
  - Contact and disscussions also with Enrico Guiraud and Danilo Piparo

ROOT-tuple Case

- The plan is to provide a ROOT file as output of the reconstruction following the structure present in prod2ntu
- This output would replace the datarec/dst
  - We need to fix the level of reconstruction to be implemented in the output
  - T0 corrections for neutral and kaon?
  - Kcharge retracking?
- The present solution is to create the ROOT structure in C++ and wrap the code around the FORTRAN to provide the output directly from A_C
- Summer Student Luigi Berducci is already working in the code development (see next slide)

- An alternative way to save the data was discussed with the ROOT v6 developers Enrico G. and Danilo Piparo
  - They are responsible for the RDataFrame and RDataSource code in ROOT v6
  - There is already the same type of work done for other collaborations like ATLAS and CMS
  - RdataSource would be responsible of reading the YBOS and parsing to a ROOT struct
  - They have offered help in case we decide to take this solution

ROOT-tuple Case Status

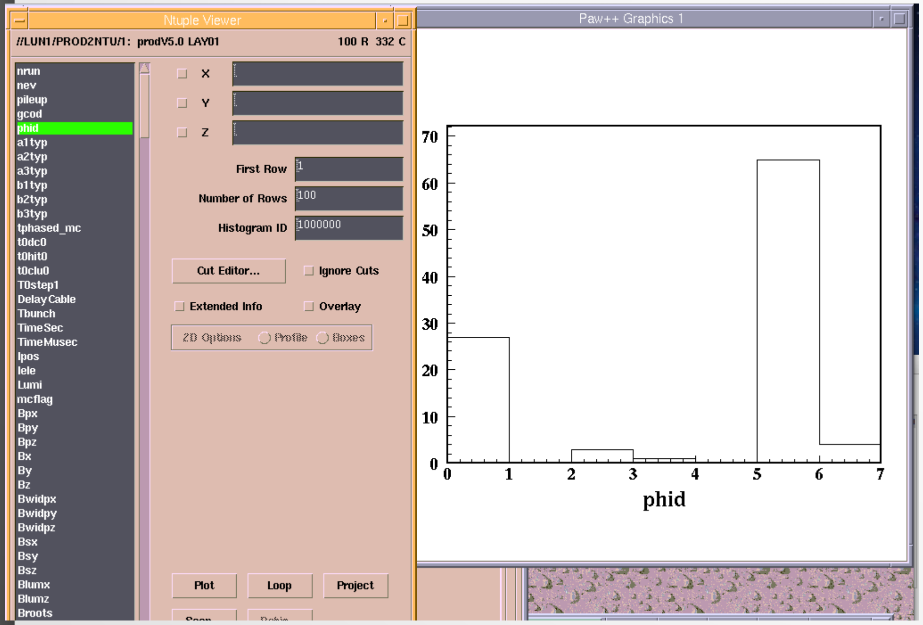
- We have already started to develop the code in C++ to parse the structure from FORTRAN to ROOT directly from A_C
- The module wrapping the C++ code is PROD2ROOT.KLOE
- We are reproducing the prod2ntu structure and adding all the banks to a TTree in a root file
- Code has been tested in fibm11 and fibm15 successfully
  - Compilation tested with two xlC versions
- Summer Student Luigi Berducci is now working on the Fort2C.cpp code in charge of creating the TTree and the TBranches with prod2ntu like variables
- All the code is hod in a repository and test to check the output compared to that of prod2ntu together with code tests to be done automatically together with the reconstruction and directly from A_C are in development
- First output can be seen in the next slide
- Communication between the C++ and FORTRAN is the most cumbersome part:
  - Need of C-struct + FORTRAN common combinations for ALL banks

First ROOT-tuple from A_C

Comparison of HBOOK and ROOT output

HBOOK/FORTRAN

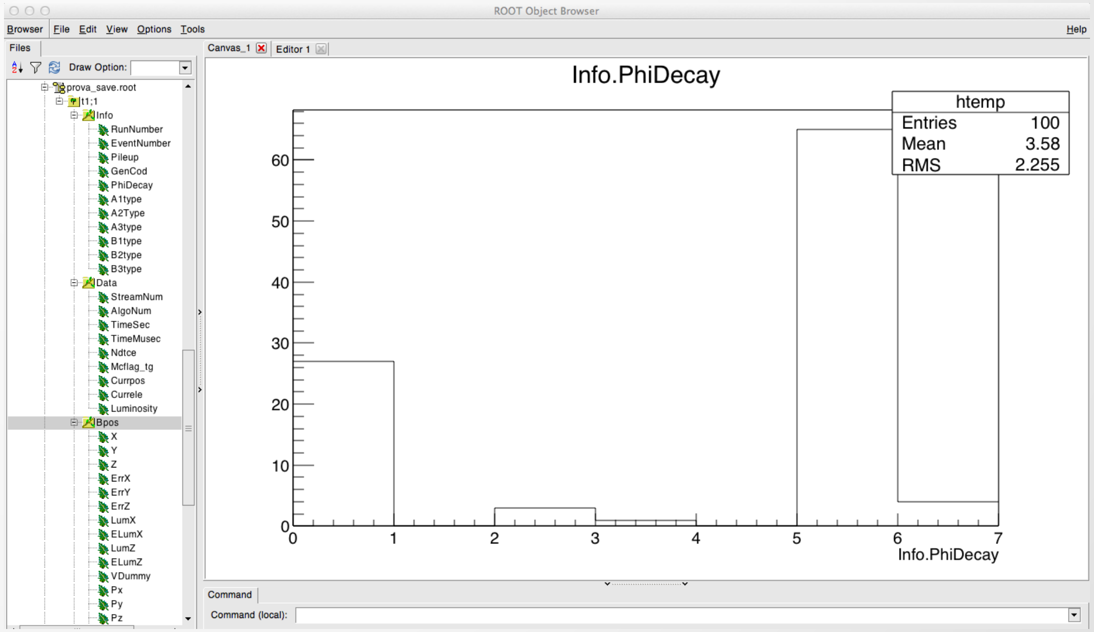
ROOT/C++



First Row 1

Number of Rows 100

Histogram ID 1000000



htemp	
Entries	100
Mean	3.58
RMS	2.255

Alternatives to our case

- There is room to improve the solution and also to change the approach by creating directly the ROOT file from the YBOS:
  - Advantages:
    - we don’t need to reproduce all structures by duplicate
    - No need of common
    - No need of FORTRAN – all done directly in ROOT or only FORTRAN wrapped in C++ to use BLOCAT
    - The experts have offered to use to create the Source code to read the YBOS
  - Issues:
    - We need ROOT v6 to use RdataFrame
    - There is no documentation on all the banks of datarec → we need to navigate the code to understand
    - For the ONLY ROOT version we need to understand the structure of the YBOS array

Roadmap

- All banks are progressively added to the ROOT-tuple
- Tests are being developed to be used by the data reconstruction
- The current code replaces the datarec output not yet tackled the dst level
- Code is stand alone: needs to be tested in the reconstruction
- Goal is to provide ROOT ntuple together with datarecs for users to test and work on the ROOT output