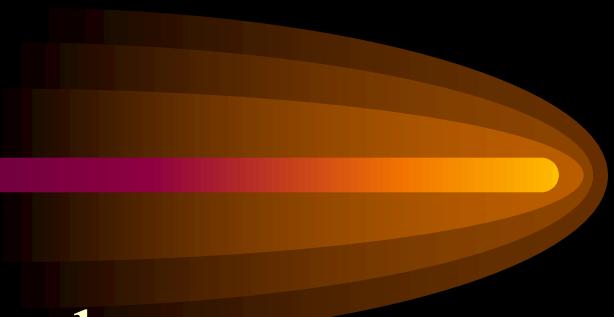
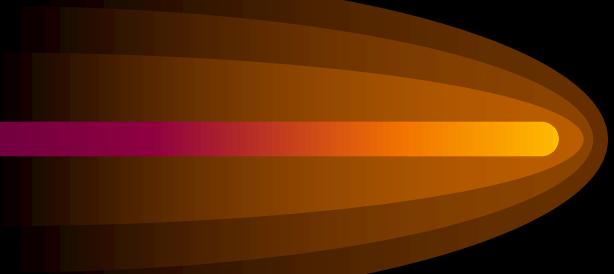


g4tools @ JLab

diff –u « since Lund »



As a reminder...

- 
- g4tools is an automatic extraction of some code found in the softinex/inlib and namespaced “g4tools” for an embedding in Geant4.
 - Pure header code. Highly portable (including iOS and Android). Easily embeddable (no “config.h” or specific build tool in the way).
 - Strongly OO. No implicit management.
 - Thread safe (no writable statics).
 - See <https://gbarrand.github.io>

What's new



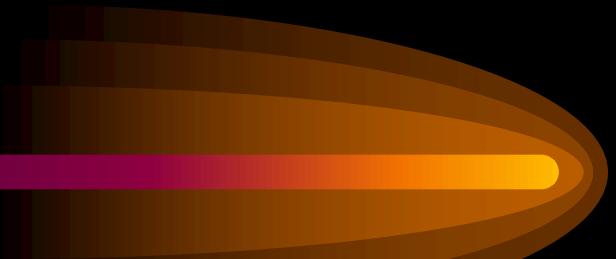
- Solved the ntuple column wise “row/event mode” problem for the ROOT format in MT and MPI seen last year.
- Prepare the ground for interactive plotting, following the vis mini-workshop at Hebden Bridge in June.

The column wise problem



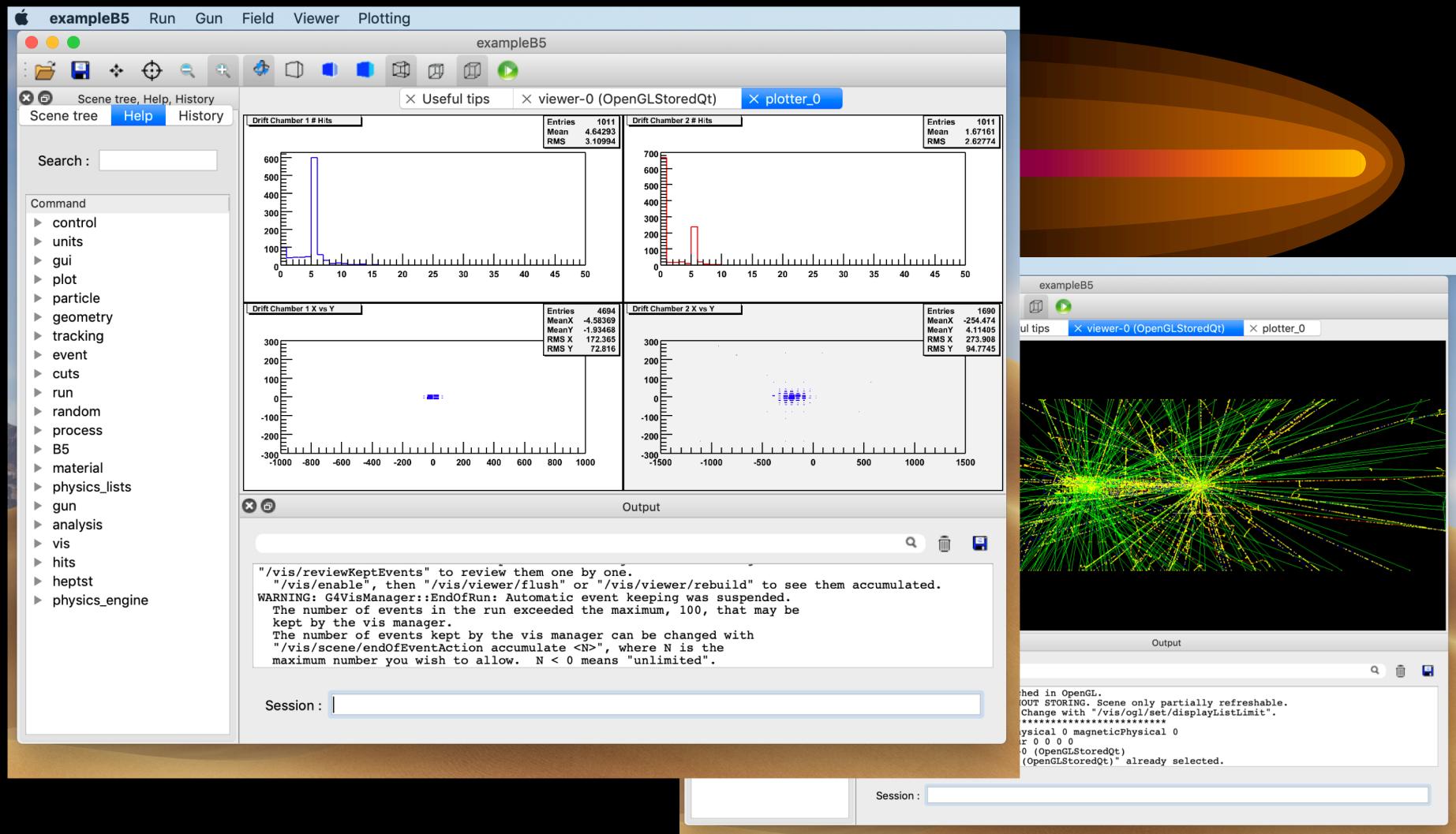
- Some people want to use, or “see”, the ntuple as a simple event model storage, even in column wise. Not a problem in single thread, but it needs an extra “basket logic” in MT or MPI.
- In MT and MPI, in each worker it is needed to stage baskets, coming from each branch, containing a fixed number of events and send them at once to the main worker handling the file.
- Done that for MT and MPI. It can go in the December release.
- But, as this mechanism uses more memory, we keep anyway the “raw column wise” mode, for people interested in “branch histogramming” only.

Prepare interactive plotting (1)



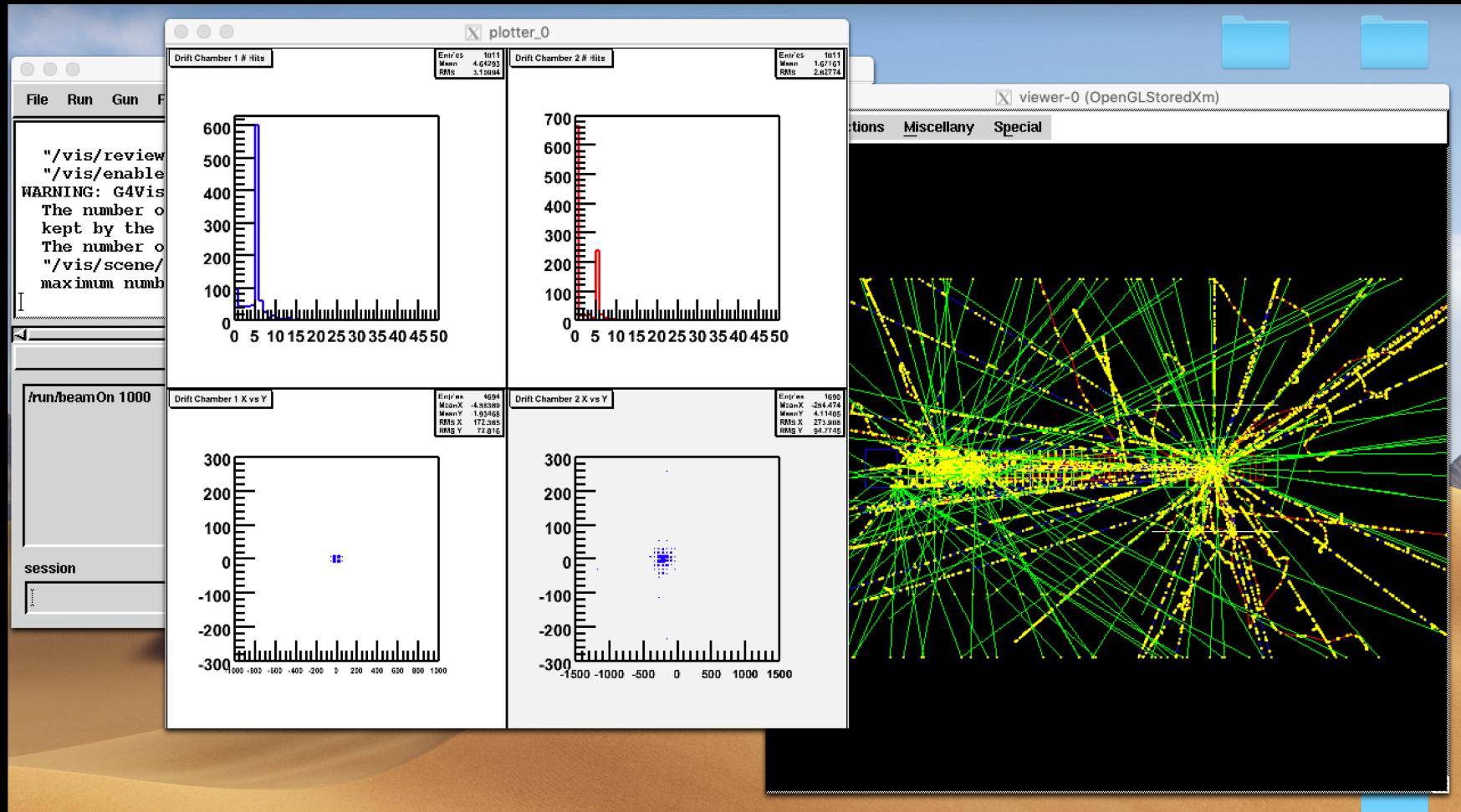
- Vis mini-workshop at Hebden Bridge in June (thanks John for organizing it).
- Agreement to (attempt) to have interactive plotting.
- All the material is here in softinex (for long), knowing that the core logic of plotting is already here in g4tools for batch plotting.
- More a surgical architectural problem to know where and how to plug it in the today interactive environments existing, in particular, for visualization.
- A good prototype, working with Qt and the “good old Motif”, done during summer.

Prepare interactive plotting (2) / Qt



Within Qt environment : GUI, commands, vis and plotting working together ☺

Prepare interactive plotting (3) / Xm



Within Xm environment : GUI, commands, vis and plotting working together ☺

Sketch of the surgery...



- Plug things not in the vis system but in the interfaces category (as decided on the Bridge).
- G4VPlottingSession : connection g4tools/plotting with G4/interfaces. Inherited by G4VBasicShell.
- A virtual G4VPlottingSession::AddPlotter() method must be implemented by the G4UIs (G4UIQt, G4UIXm) to give an OpenGL widget to g4tools to draw on screen.
- Connection with the G4AnalysisManager (that handles the histos) done by a high level template (in G4PlotHisto.hh) in an example/user code. (It permits to keep the G4AnalysisManager virgin of relationship toward any GUI and OpenGL code).
- Surgery code within the cpp “G4UI_[BUILD,VIS]_PLOTTING” macros.
- New commands :

/gui/addPlotter <plotter_name>

/plot/h1, h2 <hid> <plotter_name>

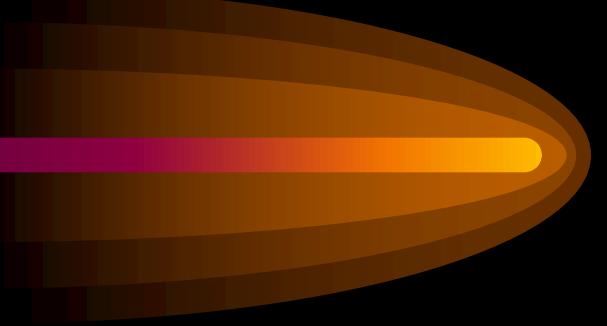
/plot/setLayout, setStyle, print # to customize plots and do also paper printing.

Prepare interactive plotting (5)



- Probably too early for December since it involves indirectly two categories (analysis and interfaces), and modifications in the build system (to handle the G4UI_[BUILD,USE]_PLOTTING, and arrange so that the GUIs/OpenGL options are available for the plotting).
- Still in my repositories, I have to cope with the new git logic... ☺
- Not so clear for a time line, but having it as a beta in the Geant4 repos could be done for next Summer (perhaps more early, depending also of other people).
- For December, I can have a version of g4tools, with all the backbone files doing the connection with Qt/Motif/OpenGL.

Conclusions



- Sort out the MT/MPI ntuple column-wise “per event” issue for the ROOT format. Ok for December.
- A way is open to include plotting within the today Geant4 interactive environments. Better to say for next year...