

HistPresent - Easy interactive analysis within the ROOT framework

O. Schaile

University of Munich, D-85748 Garching, Germany

email: Otto.Schaile@LMU.de

Presented at the ROOT2005 WORKSHOP, CERN, 26.Sept.05

INTRODUCTION

HistPresent was developed within a project for online and offline data analysis, MARaBOU, to provide access to ROOT objects like histograms, graphs and ntuples as typically used in accelerator experiments. Special emphasis is placed on the simplicity of the user interface to enable unexperienced users to display histograms and do simple operations like zooming and signal/background fitting. At the same time the full functionality of ROOT is made available like e.g. fitting of slices with complicated functions by providing templates for macros. Some extra features like simultaneous projections along X and Y and rotations of 2-dim histograms are built in.

HistPresent also exploits the interactive graphics capabilities of ROOT. In addition to ROOT's built-in graphics primitives like graphs, paves, curlylines etc. it allows interactive insertion of histograms, images, functions, xsplines and text files containing LaTeX code. Features for editing common objects on a grid are provided. Compound objects may be defined, extracted, stored in a file and reinserted with new scale and angle.

THE MARABOU PROJECT

MARaBOU [1] is a framework for online data acquisition and as offline analysis for nuclear physics experiments. It was developed at the Tandem Laboratory in Garching operated by the University (LMU) and Technical University (TUM) Munich. Design principle of MARaBOU is that the user only sees and communicates with a ROOT based system running on Linux whereas the hardware of the experiment is housed in CAMAC and VME controlled by processors running Lynx OS on VME processors.

Fig. 1 gives an overview on the dataflow within the system.

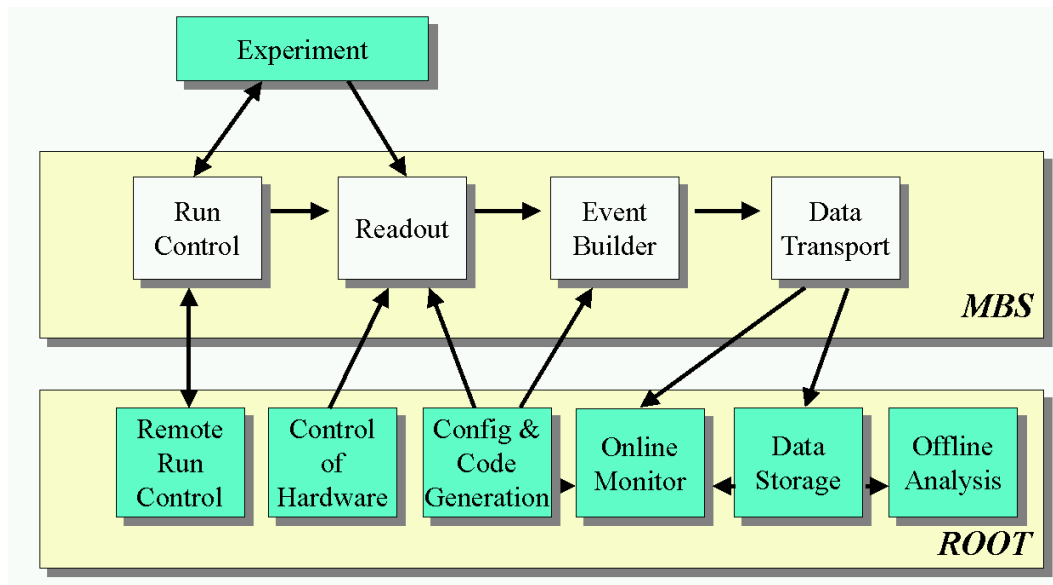


Figure 1: Marabou Tasks

DISPLAY HISTOGRAMS

Fig. 2 shows HistPresent's startup window, a file list, a list of a file contents and a widget displayed for a NTuple. All these widgets are implemented using TButton objects. Pressing the buttons triggers the obvious actions like showing the content of a ROOT file or displaying a histogram. The grey button left of a name is used as a selection. It turns green if it is active. This is used in operations like Show all selected histos or with the arithmetics operators. E.g. Selecting hpx and hpy2 then pressing + would addup histograms hpx and hpy2 resulting in a histogram hpx_plus_hpy2 which is displayed and added to the objects in memory.

Showing values of a Tree can be done in several ways:

- Just pressing a name fills and shows a 1-dim histogram
- Selecting 2 or 3 names and pressing `Show leaves` fills and shows a 2 or 3-dim histogram
- With `Edit expression` any legal expression may be formed. The input widget used for that implements a history (see fig. 2) and text completion (tab key) for variable names and frequently used functions like `TMath::Sqrt`

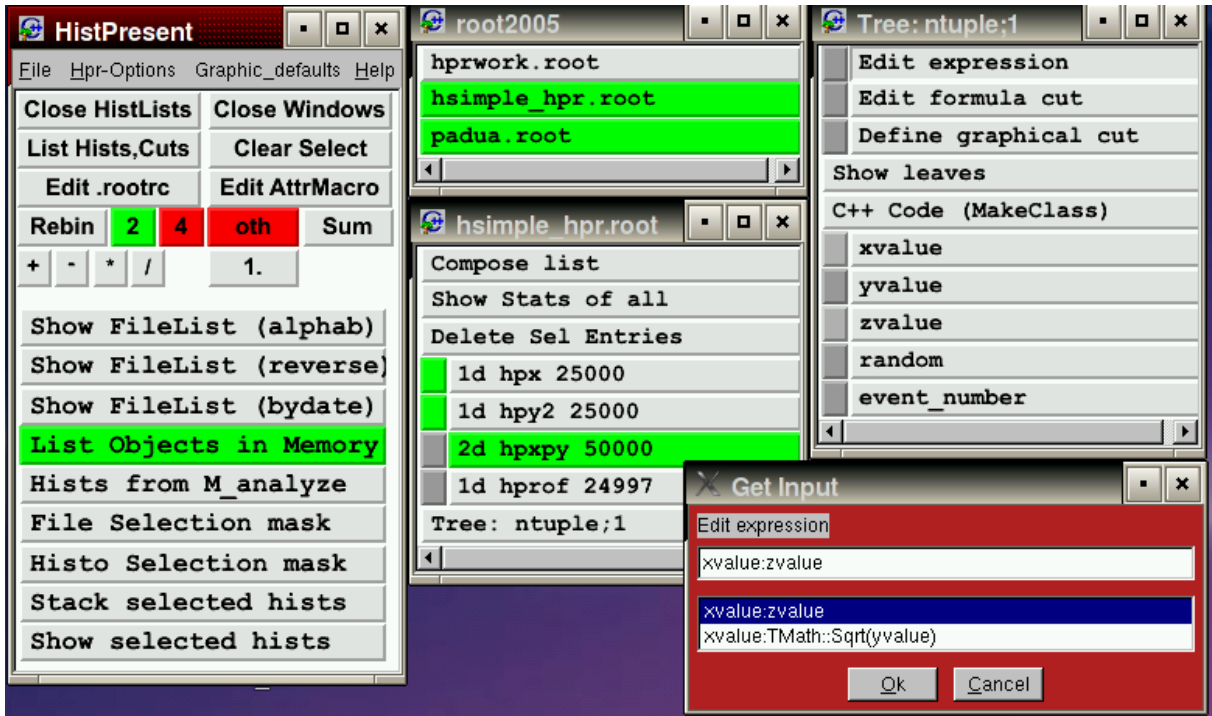


Figure 2: A sample session

A histogram is displayed in a special canvas inheriting from TCanvas with extra popup menus providing actions like `Expand`, `Superimpose`, `Projections` etc. Note: Up to now all operations only require the left mouse button whereas the standard ROOT actions from context dependent popup menus triggered by the right mouse button are still available.

The following picture shows examples: `Show selected hists` in same range, and a 2-dim histogram with a graphical cut (TCutG) applied projected on both axes.

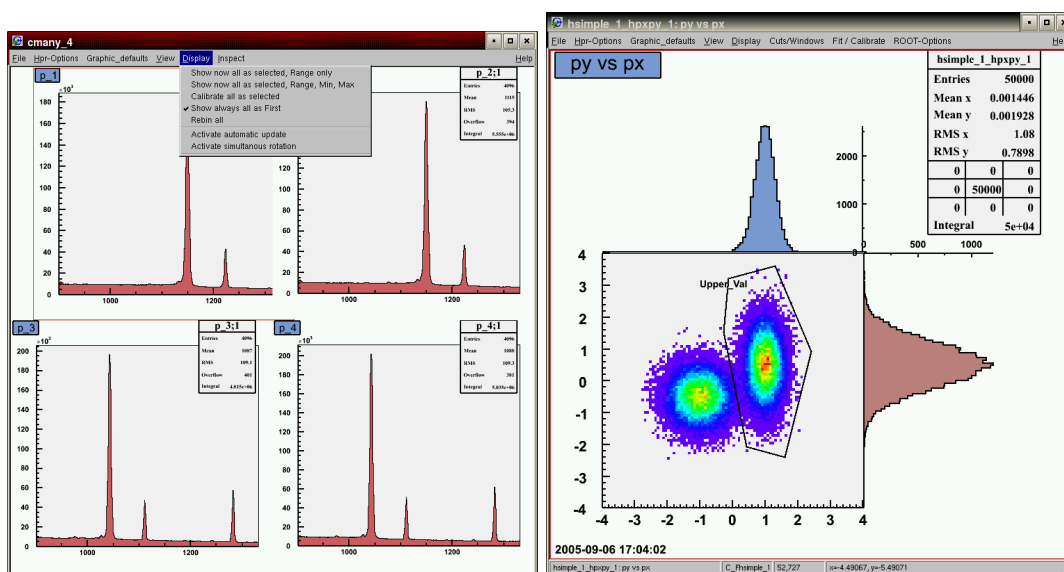


Figure 3: Show selected hists, Display_both option with graphical cut

FITTING FUNCTIONS

Among others HistPresent's canvas provides a popup menu **Fit/Calibrate** (see fig. 3). Several predefined functions like **Polynomials** and **Gauss+High Tail** are available. In this case marks which are activated with the middle mouse button are used to define limits and approximate means for a fit. Start parameters are estimated and editable by the user.

A second more general way is to use a ROOT macro. Commands **Edit User Fit Macro** and **Execute User Fit Macro** are provided for this. If with **Edit User Fit Macro** a non existing file name is given the user can choose from a few examples as templates. Again this is to put the entrance threshold to use ROOT as low as possible.

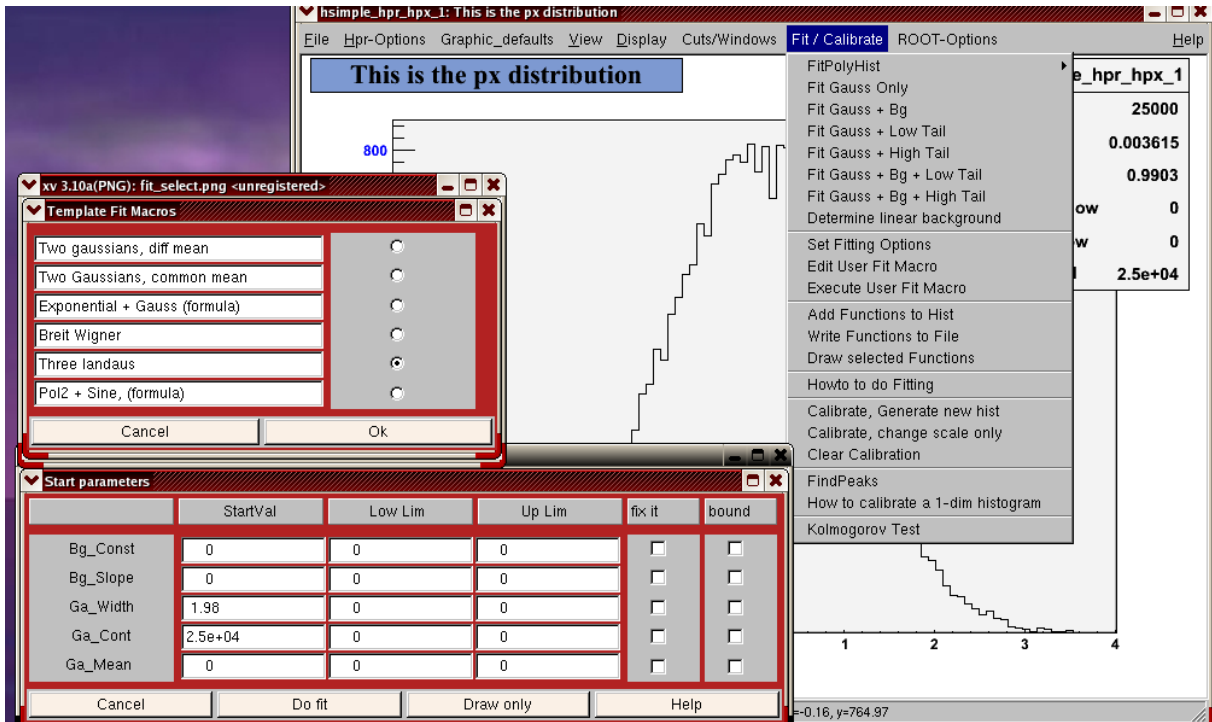


Figure 4: Widgets used with Fitting

For 2-dim histograms hooks are provided to fit slices and 2d functions. Also here examples are provided to start with.

Fig. 5 shows the 2-dim histogram from the builtin tutorial and the result of the example fit macro.

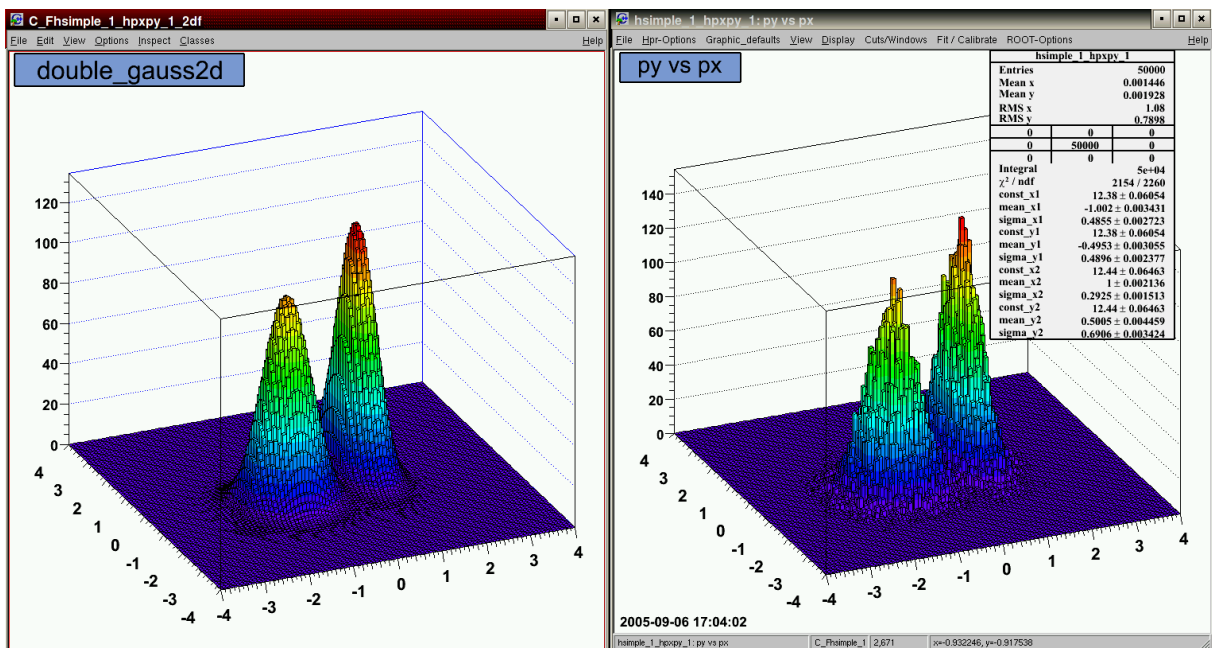


Figure 5: Example of 2d fit

THE GRAPHICS EDITOR

A graphics editor canvas is invoked from the **File** popup in the startup window. For convenience three sizes are predefined: A4-portrait, A4-landscape and user. In the canvas an edit grid is provided. This is implemented by overloading `TCanvas::HandleInput` by forcing mouse click coordinates on the user defined grid. To overcome the problem of the finite pixel resolution a method is provided to align all objects (selectable by type) in the canvas at the grid. Global editing facilities (e.g. change color of all `CurlyLines`, **Delete enclosed objects** and zooming are available.

In addition to ROOT's builtin graphics primitives like `TArcs`, `TCurlyLines` the following features / objects are accessible by a menu (see fig. 7):

- **Insert images (TASImage), histograms, graphs**. The user has to define a Pad as container for the object.
- **Insert a function**. After giving the number of parameters a widget (fig. 6) allows input of the needed values.
- **Insert Text / Latex from file or keyboard**. This widget facilitates input of Text / LaTeX by implementing input from and providing a history facility for input from keyboard. Text attributes may be adjusted from the widget (see fig. 6). Optionally a filter which tries to convert standard LaTeX to ROOT's Latex may be switched on.
- **Insert compound object**. Compound objects may be defined by enclosing graphic elements by a `TCutG` and saved on file. They can be reinserted applying a scale and rotation.
- **Draw a XSpline**. XSplines [2] as used e.g. in XFig [3] are very simple to control by moving the controlpoints or changing the shapefactors between -1 and 1. `HistPresent` provides a Control Graph Mixer for that. Parallel graphs, optionally with railway like filling, may be added.

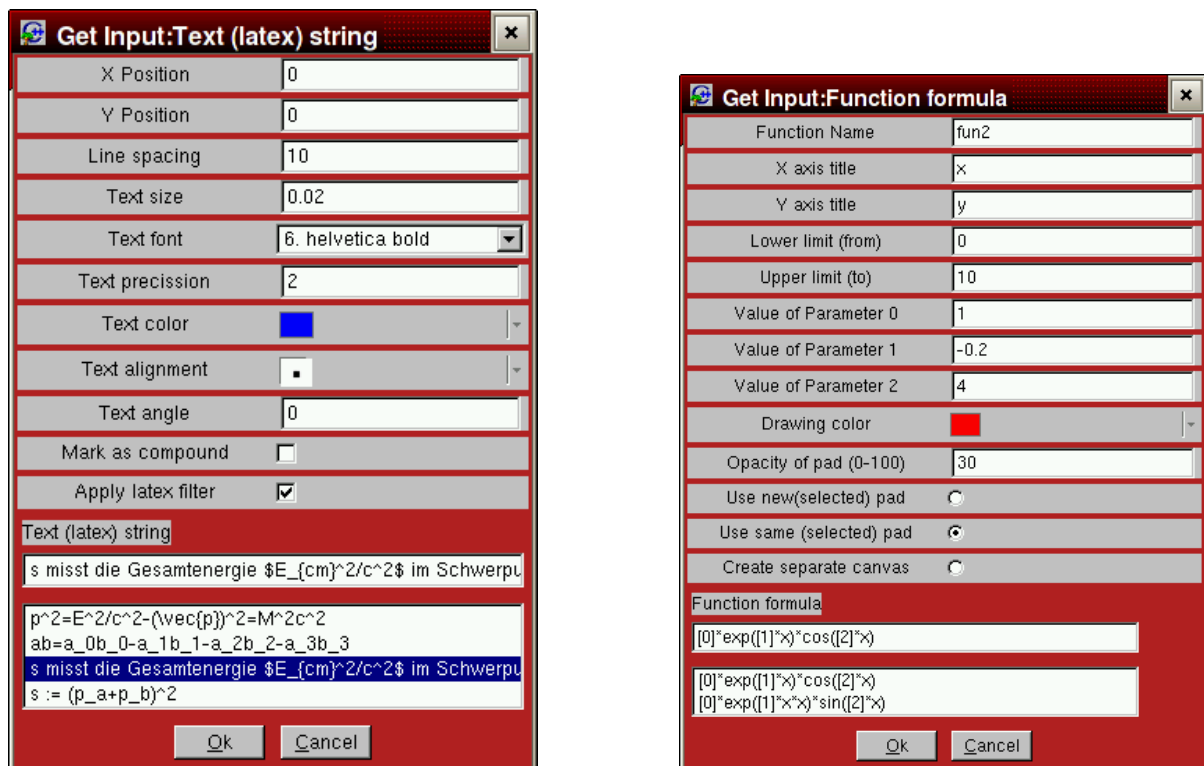


Figure 6: Input Latex, Input function

DOCUMENTATION, AVAILABILITY, DOWNLOAD

The documentation of `HistPresent` can be found here:

<http://www.b1.physik.uni-muenchen.de/marabou/html/hpr>

`HistPresent` is part of `MARaBOUs` CVS repository. The structure of directories and the `Makefile` is similar to the ones of `ROOT` itself. It was only used and tested on Linux type systems.

To install `HistPresent` please consult this URL:

<http://www.b1.physik.uni-muenchen.de/marabou/html/UpdateHprFromCVS.html>

The screenshot shows the Hpr-Edit software interface. The window title is "page_1". The menu bar includes "File", "Hpr-Options", "Graphic_defaults", "View", "Hpr-Edit", and "Inspect". The toolbar contains various drawing and editing tools. On the left, there is a "Style" panel with fields for "Name" (set to "TLatex"), "Line" (set to "2"), "Text" (set to "23"), and font settings (set to "6. helvetica bold" and "11 Bottom, Left"). Below the style panel is a "Hpr Edit Co" toolbar with buttons for "Insert image (gif, jpg)", "Insert histogram", "Insert graph", "Insert Function", "Text (Latex) from file", "Text (Latex) from keyboard", "Insert compound object", "Draw an axis", "Draw a XSpline", "Define a box shaped region", "Define polygon region", "Mark as compound object", "Extract as compound obj", "Delete enclosed objects", "Modify objects by keyboard", "Zoom In", "Zoom Out", and "UnZoom".

The main editor area is a grid with the following content:

- A Feynman diagram showing an electron-positron annihilation process into two electrons and a photon.
- A photograph of a rose.
- A complex mathematical equation for the differential cross-section $\frac{d\sigma}{d\cos\theta}(e^+e^- \rightarrow f\bar{f})$.
- A 2D histogram plot.
- A line graph plot.
- The text "Examples for XSplines" followed by two curved lines, one solid blue and one dashed black, with circular control points.

Figure 7: Example of an Editor page

References

- [1] R. Lutter, O. Schaile, K. Schöffel, K. Steinberger, P. Thirolf, and C. Broude, "MARaBOU- A MBS and ROOT Based Online/Offline Utility" *Proceedings of the 11th IEEE Real Time Conference*, Santa Fe, NM, June 14-18, 1999 see also <http://www.bl.physik.uni-muenchen.de/marabou/html>
- [2] Carole BLANC and Christophe SCHLICK, "X-splines : A Spline Model Designed for the End User" *Proceedings of SIGGRAPH '95* see also: <http://dept-info.labri.fr/~blanc/abst.html#sig95>
- [3] <http://www.xfig.org/userman/>