



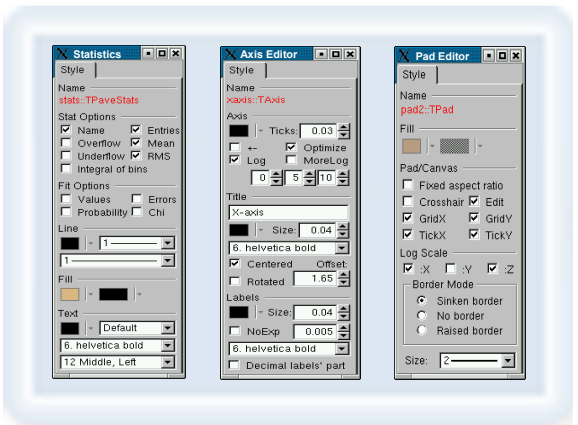
I. Antcheva <sup>1,2)</sup>, B. Bellenot <sup>3)</sup>, R. Brun <sup>1)</sup>, O. Couet <sup>1)</sup>, V. Onuchin <sup>4)</sup>, F. Rademakers <sup>1)</sup>  
<sup>1)</sup> CERN – European Organization for Nuclear Research, Geneva, Switzerland  
<sup>2)</sup> Funded by Particle Physics and Astronomy Research Council, Swindon, UK  
<sup>3)</sup> ALCAN Aluminium Valais, Chippis, Switzerland  
<sup>4)</sup> Institute of High Energy Physics, Protvino, Russia

The Graphical User Interface (GUI) design is an important component of the ROOT framework. Two sets of classes, recently introduced in ROOT v4.01, are presented in this paper: the graphics editor and the GUI builder.

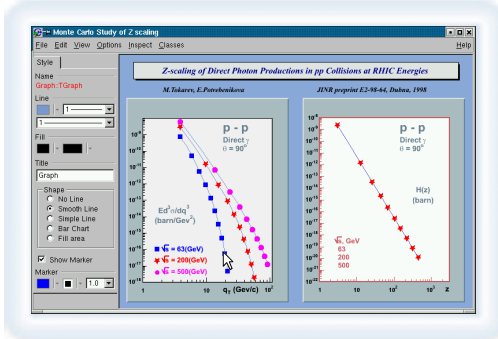
## ROOT Graphics Editor

The graphics editor is split into discrete units of so-called object editors. This makes the GUI easier to design and adapt to the users' profiles. The only convention to follow is to derive the code object editor from the `TGedFrame` base class, and to use as a name the class name concatenated with 'Editor', i.e. for `TGraph` objects the editor is the `TGraphEditor`.

## Different Object Editors



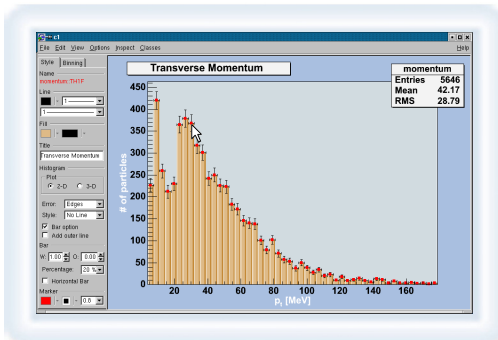
## TGraph Editor



TGX11

- They give an intuitive way to edit objects in a canvas with immediate feedback.
- The three-mouse-clicks rule of navigation limits the number of levels for completing a single task.
- The user interface gives full control to users.
- Related actions work the same way and reinforce the understanding of the functions.
- Complexity is reduced by hiding some GUI elements and revealing them when necessary.

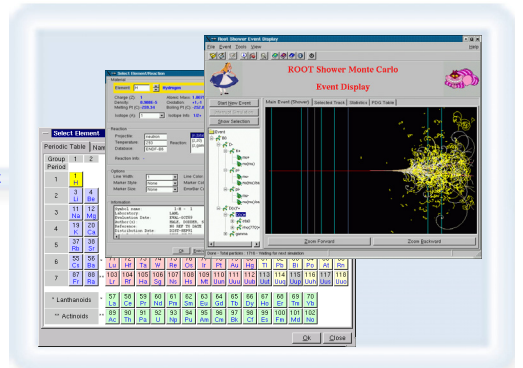
## TH1 Editor



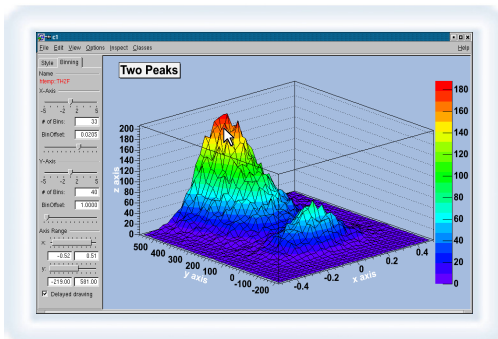
TGWin32GDK

TVirtualX

## User Applications



## TH2 Editor



TGQt

The ROOT widgets (window's gadgets) are fully cross-platform. The GUI classes interface to the platform-dependent low level graphics system via the abstract class `TVirtualX`. Concrete versions of this abstract class have been implemented for `x11`, `Win32`, and `Qt`.

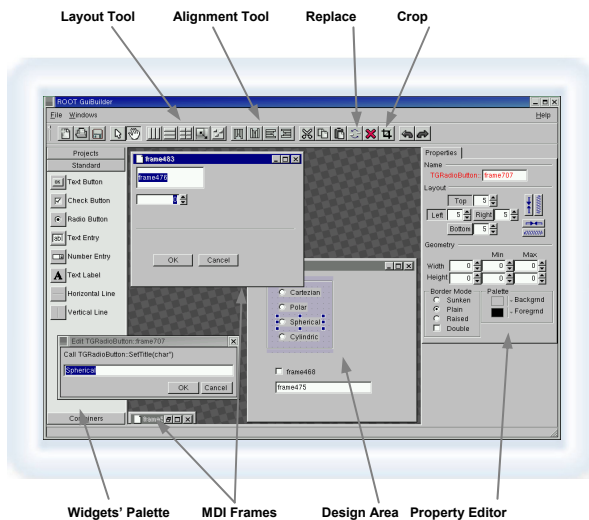
Thanks to this single graphics interface, porting to a new platform requires only the implementation of `TVirtualX`.

The benefit of applications running on more than one kind of computer is obvious - it increases the program's robustness, makes their maintenance easier and improves the reusability of the code.



## ROOT GUI Builder

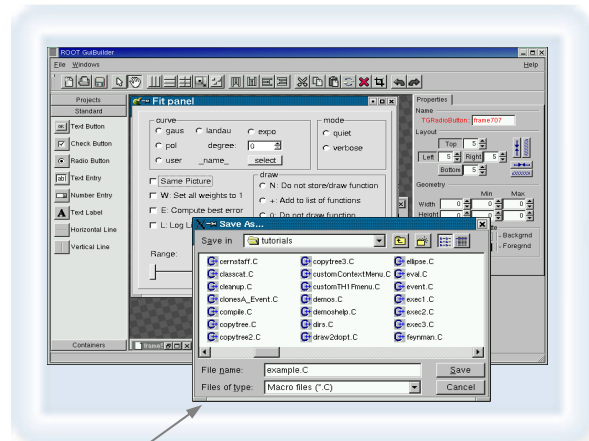
The GUI builder greatly simplifies the process of designing GUIs based on ROOT widget classes. It provides a complete tool for creating, laying out, and programming user interfaces.



The user interfaces can be assembled by dragging GUI elements from the widgets' palette and dropping them in the design areas.

Any design area can represent a main application window or a dialog window.

Widgets can be resized, grouped and moved to desired positions, and can be easily assigned to one or another parent widget. Each widget's properties can be set using the Property Editor tools of the builder.



Using **Ctrl+S** or **SaveAs** dialog, users can generate C++ code in a macro that can be edited and executed via the CINT interpreter:

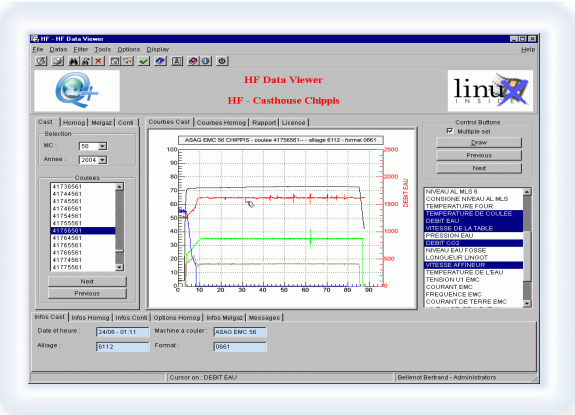
```
root[0] .x example.C
```

```
// transient frame
TTransientFrame *frame2 = new TTransientFrame(gClient->GetRoot(), 760, 590);
// group frame
TGroupFrame *frame3 = new TGroupFrame(frame2, "curve");
TRadioButton *frame4 = new TRadioButton(frame3, "gaus", 10);
frame3->AddFrame(frame4);

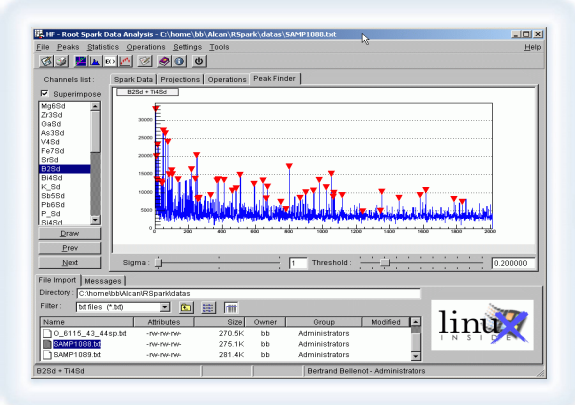
frame2->SetWindowName("Fit Panel");
frame2->MapSubwindows();
frame2->Resize(frame2->GetDefaultSize());
frame2->MapWindow();
}
```

## Application Examples

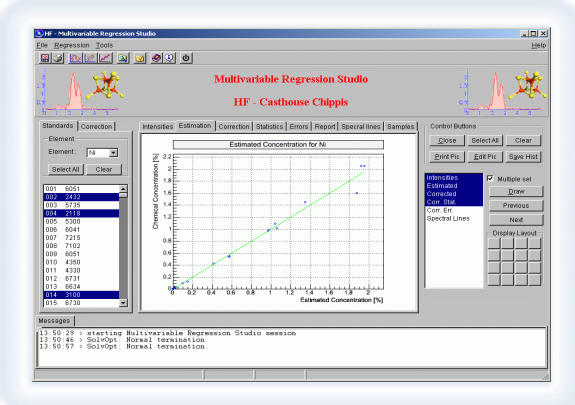
The non-HEP examples presented below of concrete applications are from ALCAN Aluminium Valais, Chippis, Switzerland.



The application **HFViewer** is used for validation of the product parameters before the expedition. It is cast-house processes data viewer.



The **RSpark** is a pick finder application in to a series of spark intensities that are taken from an emission spectrometer in the aluminum cast-house



The Multivariable Regression Studio is based on the basic concepts of the correction methods and data processing.

For more information see: <http://root.cern.ch>  
For any questions please use following address: [rootdev@pcroot.cern.ch](mailto:rootdev@pcroot.cern.ch)