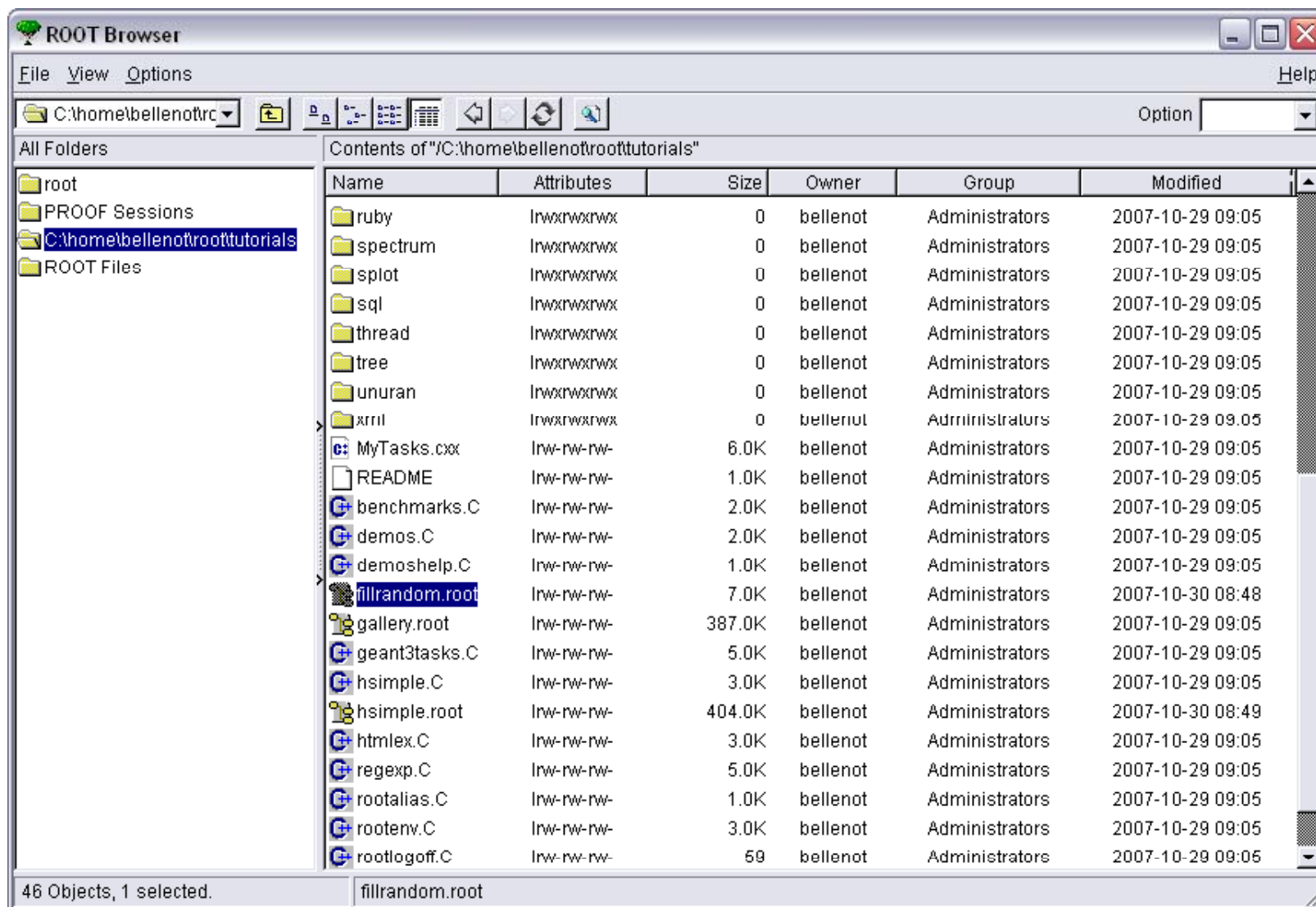


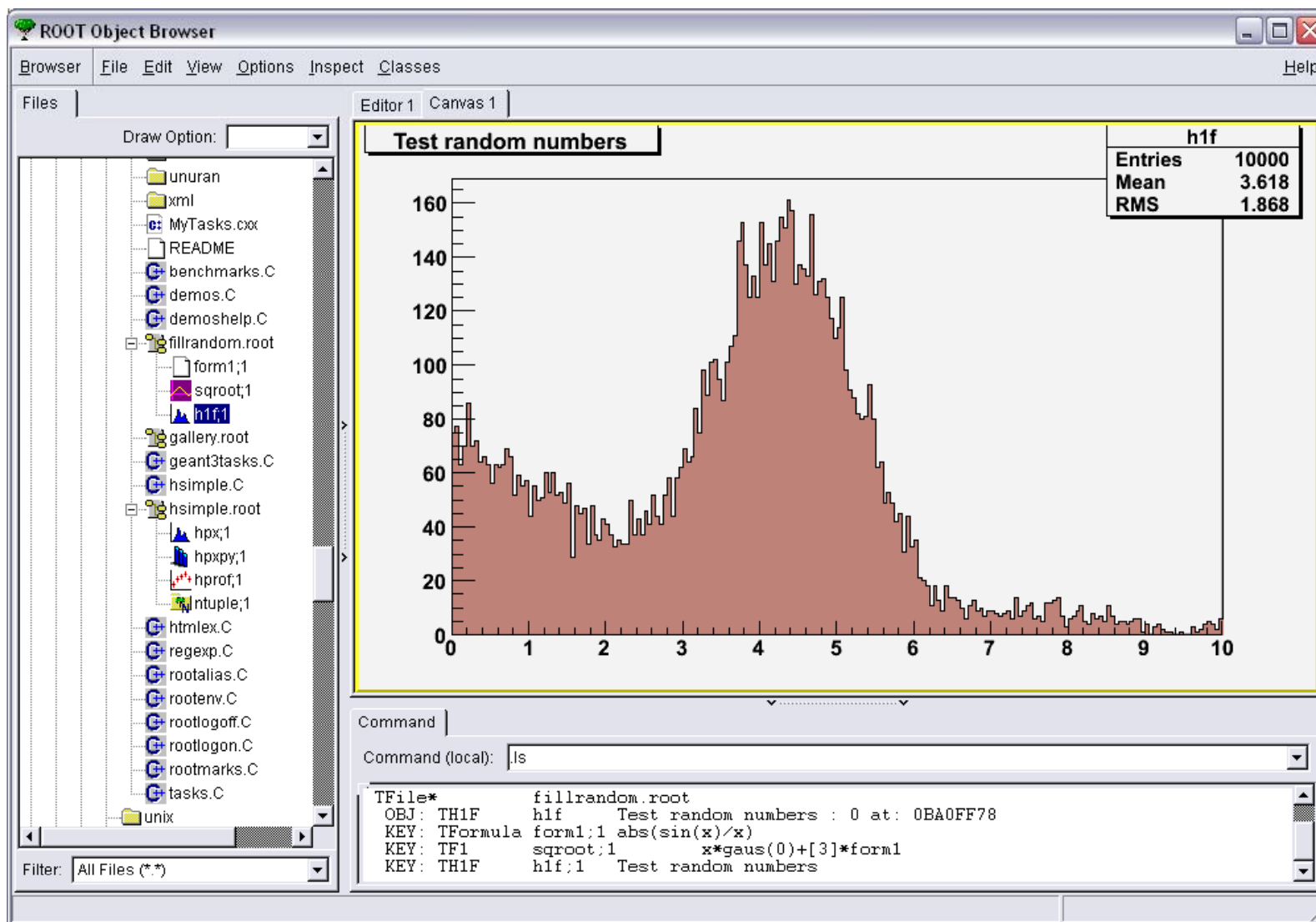
The new dynamically and extensively customizable ROOT browser

Bertrand Bellenot



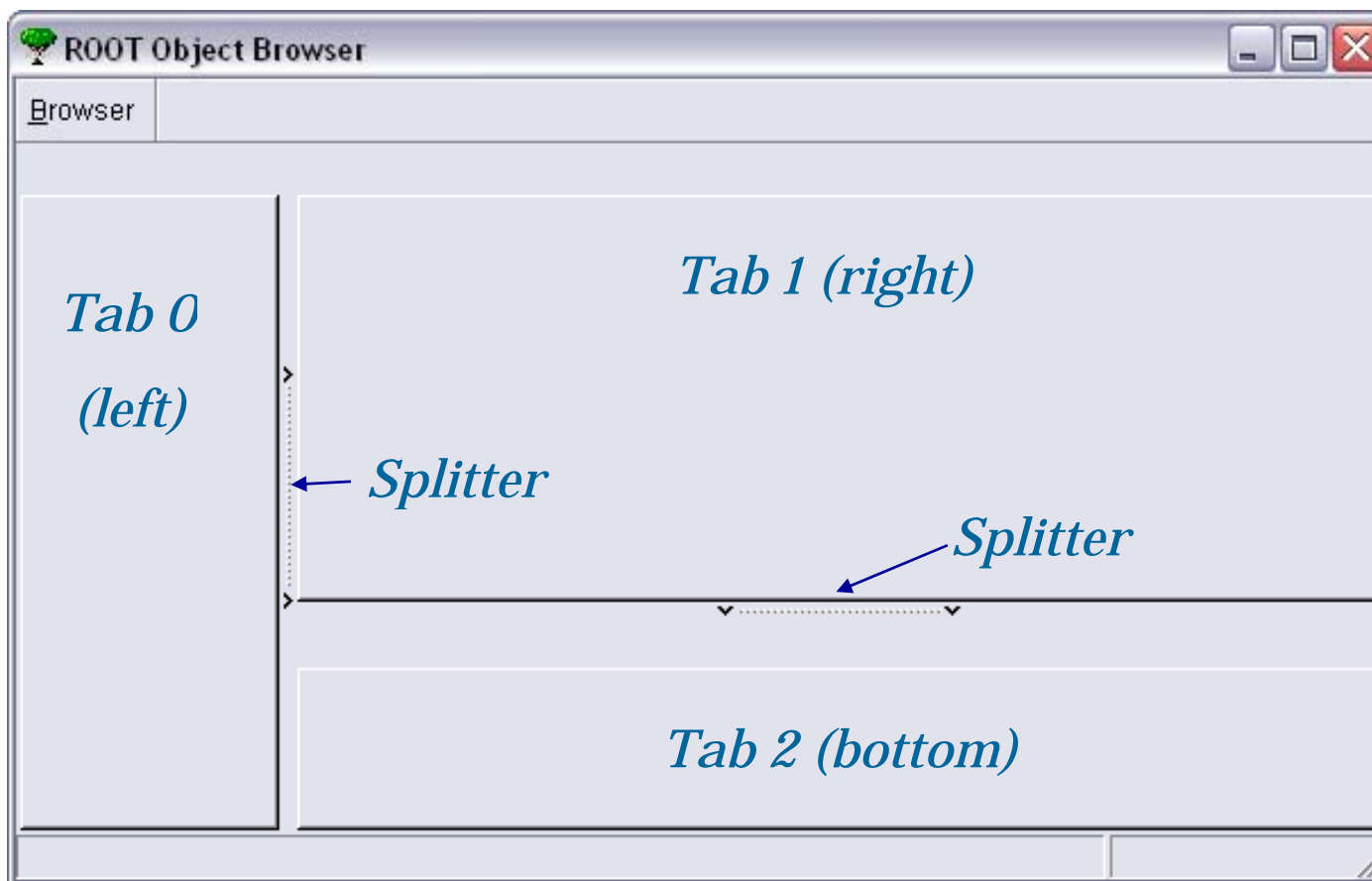
From this...





- Features
 - New layout
 - Plug-in management
 - Available (built-in) plug-ins
 - File browser
 - Canvas
 - Text editor
 - Command redirection
 - HTML browser
 - ...

The new browser has three main tabs, separated by horizontal and vertical splitters:



Each Tab can be populated with plug-ins. A plug-in can be a macro or a command, as shown below:

Example 1 - create a browser and execute a macro as plugin:

```
Tbrowser b;  
b.ExecPlugin("MyPlugin", "MyMacro.C");
```

In this example, the macro “MyMacro.C” creates a TGMainFrame, which will be embedded at the default location (right tab), creating a new tab element called “MyPlugin”.

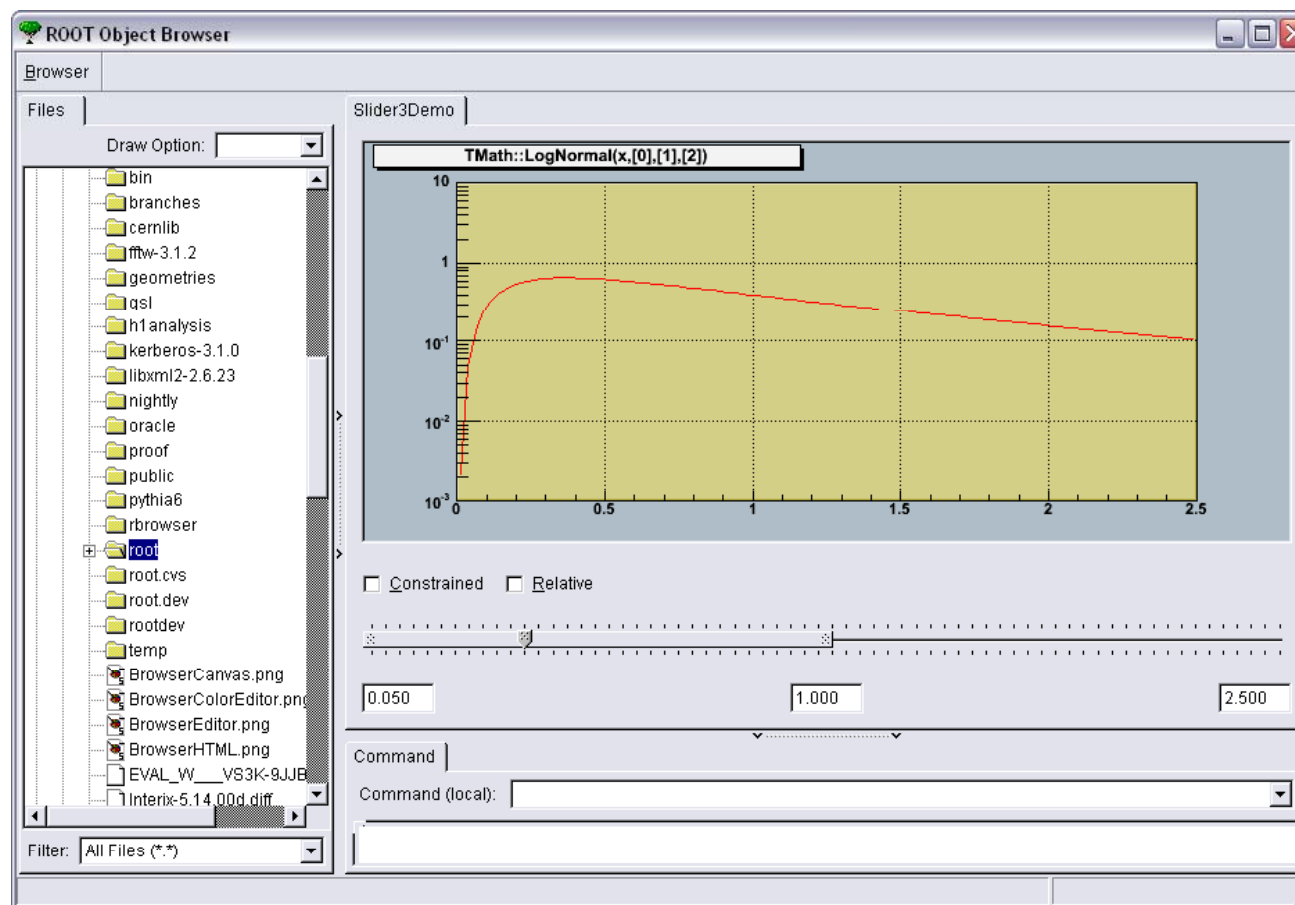
To be able to use a macro as plug-in, it must create a TGMainFrame, passing “gClient->GetRoot()” as parent window in its constructor, i.e.:

```
fMain = new TGMainFrame(gClient->GetRoot(), 300, 150);
```

Plug-in Management (cont)

Example of plug-in using the macro Slider3Demo from ROOT tutorials

```
b.ExecPlugin(0, "gui/Slider3Demo.C");
```



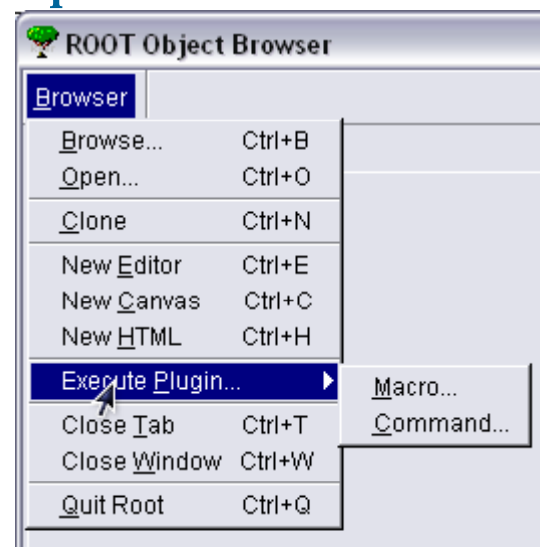
Example 2 - create a browser and execute a command as plug-in:

```
Tbrowser b;  
b.ExecPlugin("Canvas", 0, "new TCanvas()");
```

In this example, the TCanvas will be embedded at the default location (right tab), creating a new tab element called “Canvas”.

This can be achieved with any class creating (or inheriting from) a TGMainFrame. The only requirement is to pass `gClient->GetRoot()` as parent in the constructor, as shown previously.

Plug-in execution is also available from the “Browser” menu:



There are several plug-ins already available in ROOT. They are:

- File browser (**TGFileBrowser**). This is the core of the new browser, allowing to browse system directories and ROOT files and execute macros, as in the old browser.
- Text editor (**TGTextEditor**) allowing to edit, execute and compile macros. Double-clicking on a macro file in the file browser while the active tab contains a text editor will open the macro in the editor.
- Canvas (**TCanvas**). Double-clicking on a graphic object inside a ROOT file will display the histogram in the canvas.
- Command (I/O) redirection (**TGCommandPlugin**). This plug-in creates a combo box (populated with root command history) to enter interpreter commands (stdin), and a text view displaying the output (stdout/stderr).
- HTML browser (**TGHtmlBrowser**), allowing to browse web pages and to open remote ROOT files (via http protocol).
- Proof (**TSessionViewer**), allowing to manage Proof and local sessions
- GL Viewer, ROOT's OpenGL viewer.

The built-in plug-ins to be loaded by the browser at creation time are specified in the ROOT configuration file (system.rootrc or .rootrc)

In this file, user can also specify which browser has to be used:

```
Browser.Name:                TRootBrowserLite
# Browser Options (plugins)
# F: File browser E: Text Editor H: HTML browser
# C: Canvas I: I/O redirection P: Proof G: GL viewer
Browser.Options:              FECI
```

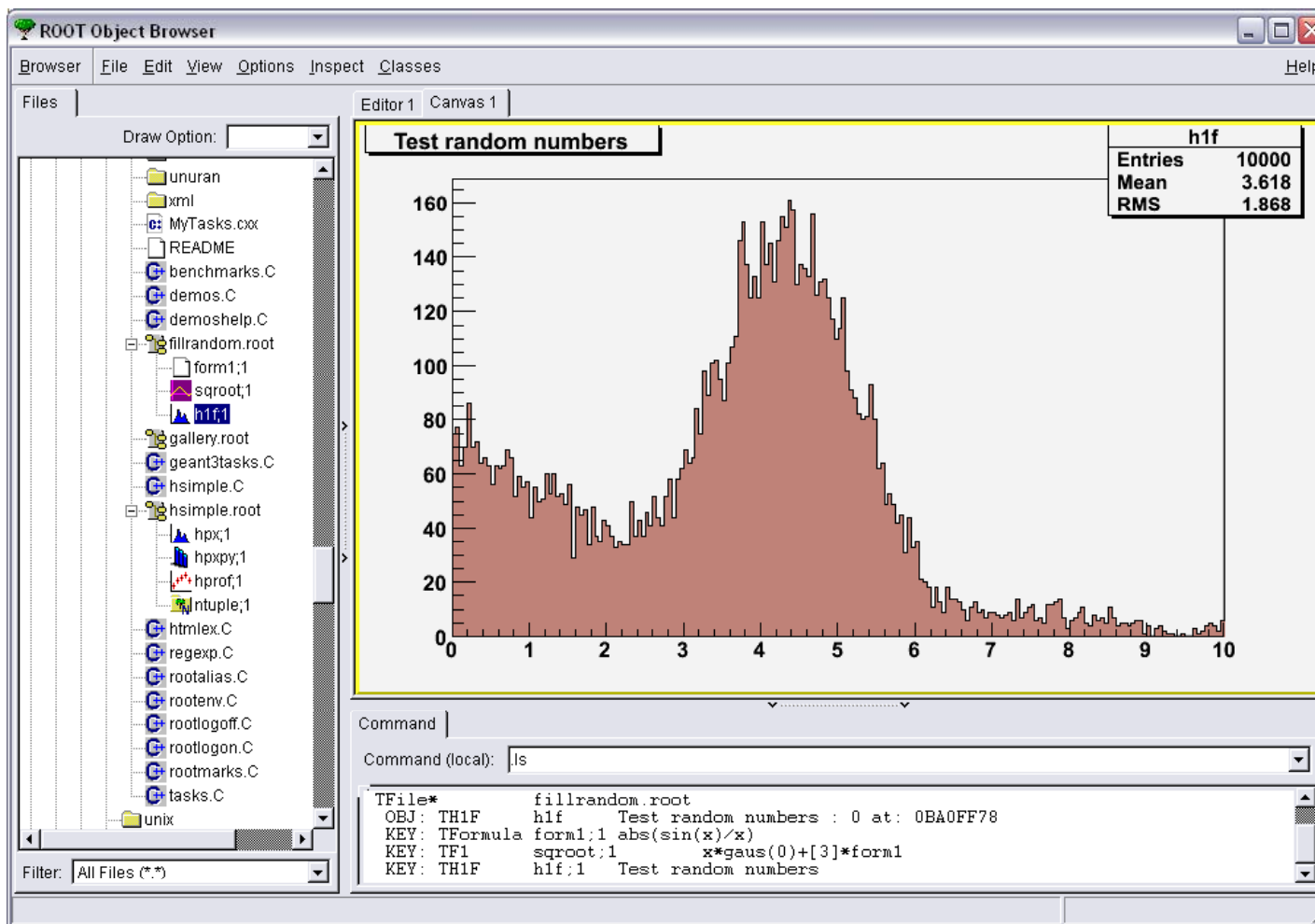
Browser.Name: specify the browser real implementation, TRootBrowser being the new browser and TRootBrowserLite being the old one

Browser.Options: plug-ins to be included at browser startup time.

It is also possible to specify (overwrite) the plug-ins option in the TBrowser constructor, i.e.

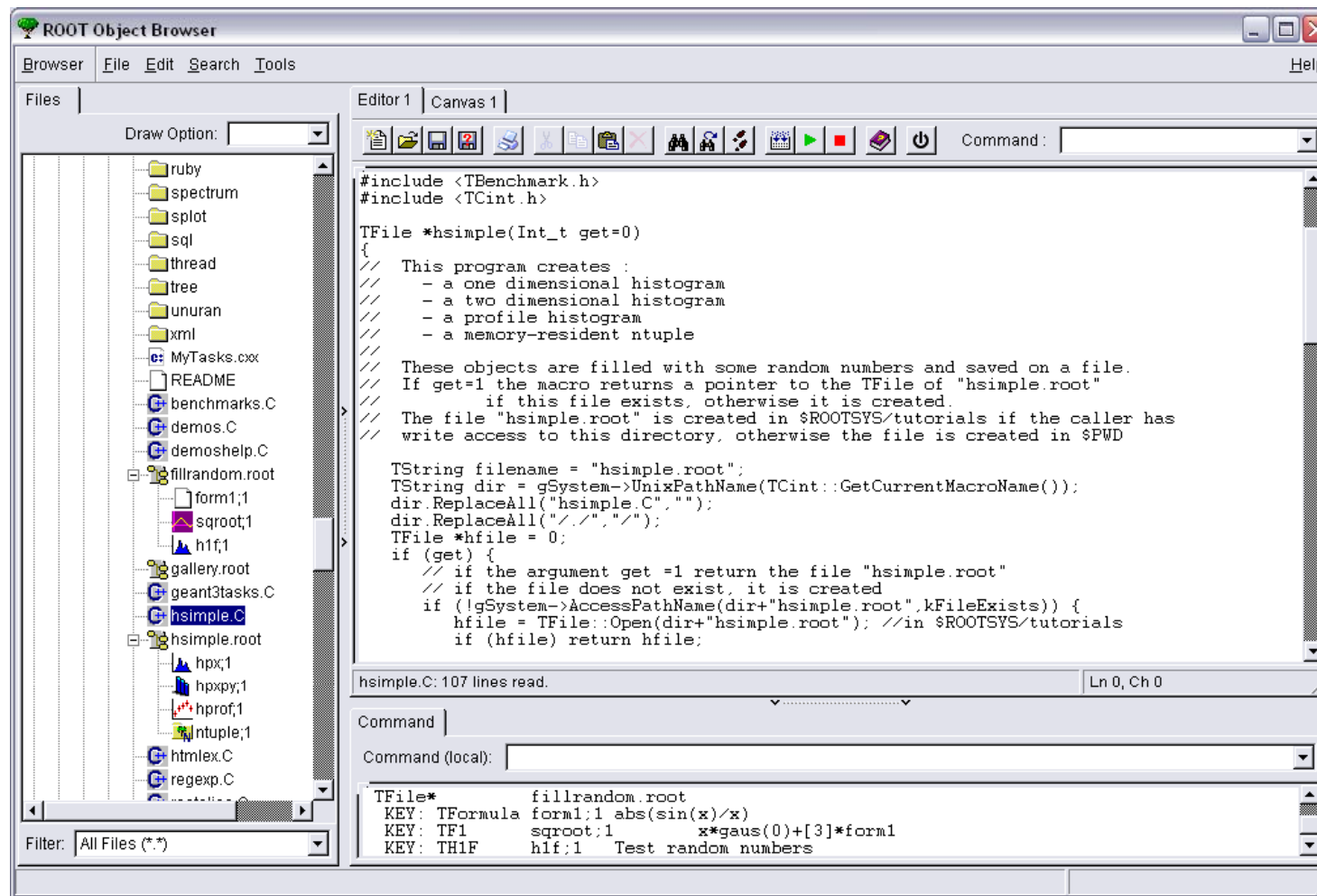
```
new TBrowser("Browser", "ROOT Browser", 0, "FEHCI");
```

File browser, canvas and command plug-ins:



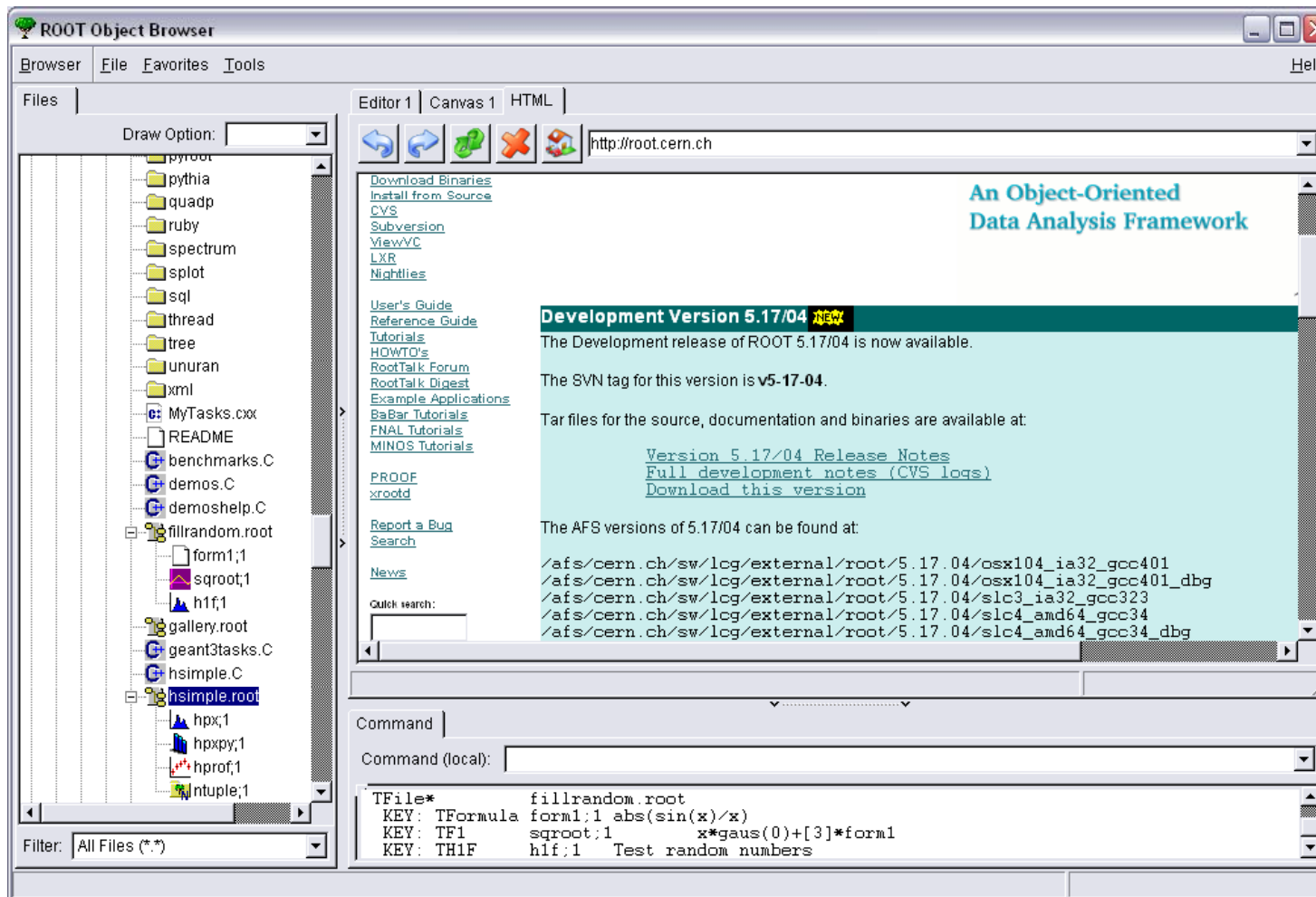
Examples (cont)

File browser, editor and command plug-ins:



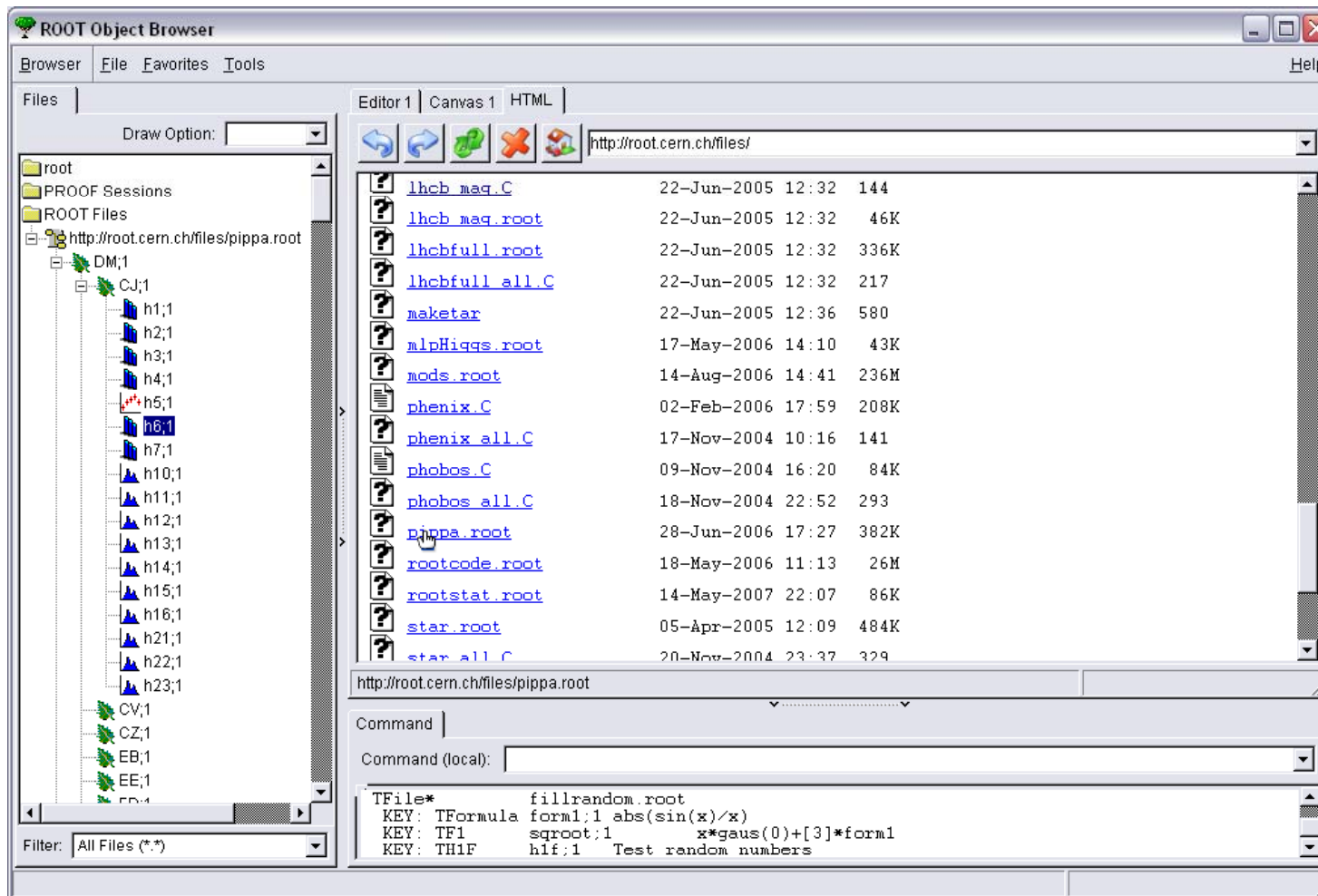
Examples (cont)

File browser, HTML and command plug-ins:



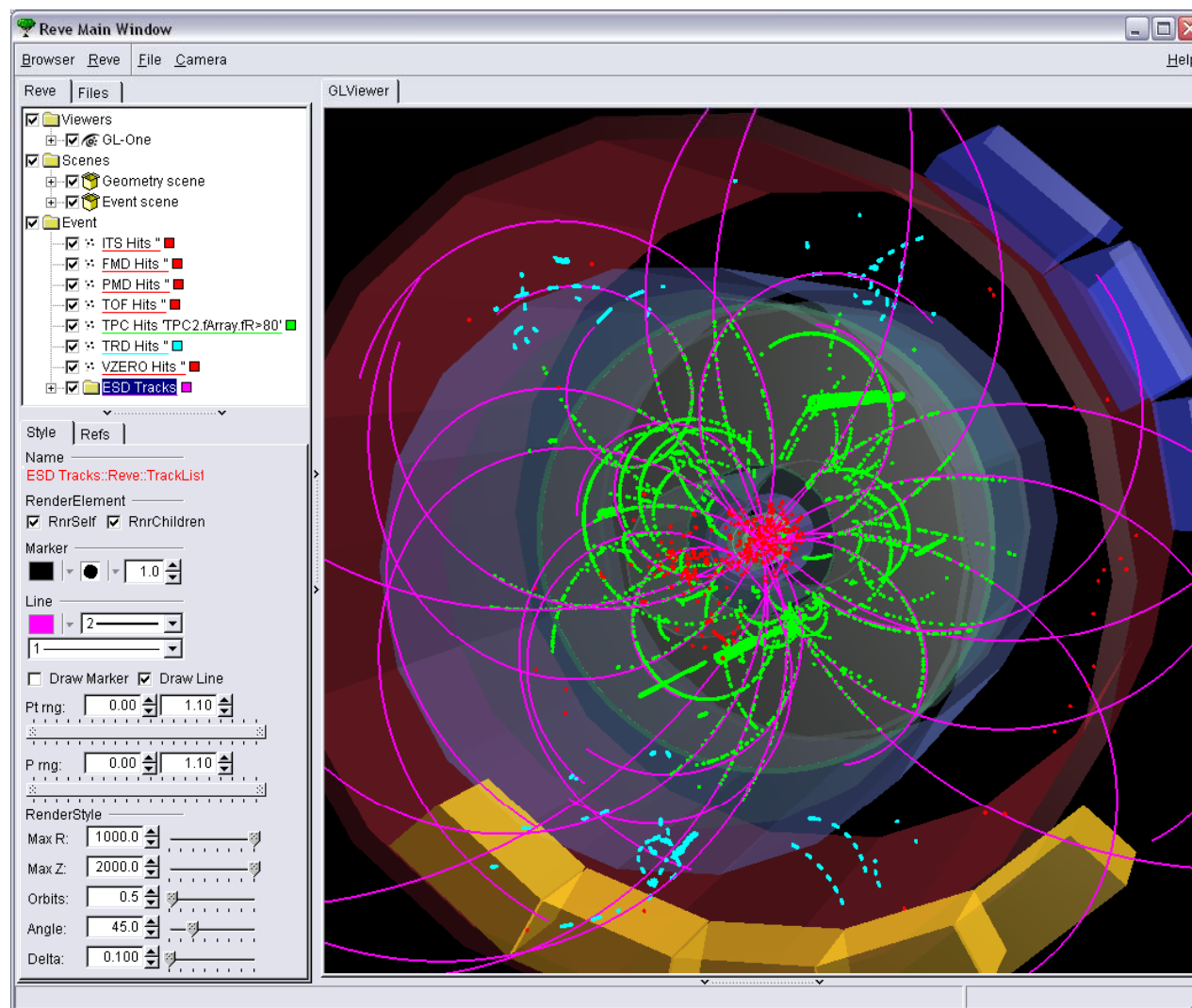
Examples (cont)

File browser, HTML and command plug-ins, browsing ROOT file via http:



Real Application Example

Alice Event Viewer (thanks to Matevz Tadel)



- The new browser provides a fully and dynamically configurable interface.
- It uses a dynamic plug-in mechanism to reduce the code size and to extend its functionality.
- Any kind of user defined plug-in can be added without too much effort from the user side.
- Already available in svn trunk (ROOT v5.17.05)

Users feedback is welcome!