JavaScript ROOT

JavaScript ROOT

Bertrand Bellenot[1] and Sergey Linev[2]

[1]CERN, PH-SFT, Geneva, Switzerland
[2]GSI, RB-EE, Darmstadt, Germany

E-mail: s.linev@gsi.de

Abstract. The redesign of JSRootIO code made it modular and usable in other projects. Many new interactive features are provided. JavaScript ROOT also implements user interface for *THttpServer* class.

1. Introduction

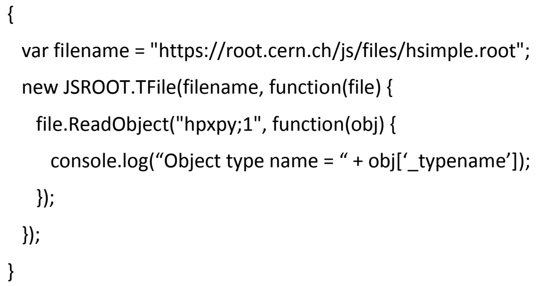
ROOT is used by almost all experiments throughout high energy and nuclear physics to write, read and analyze data. In order to allow browsing (inspecting) ROOT files on the web, a JavaScript version of the ROOT I/O subsystem has been developed, JSRootIO [1]. It provides possibility to read objects data from binary ROOT files and display them with SVG-based graphics in web browsers.

2. Code redesign

Different functional parts of JSRootIO were closely coupled together making it difficult to use them in other projects or to extend their functionality. Therefore code has been redesigned, trying to clearly separate different modules from each other. The new project obtains the name JavaScript ROOT.

2.1. ROOT files reading

To deserialize objects data from binary buffers “streamer infos” are used in JavaScript. Usage of new *JSROOT.TBuffer* class makes the logic of JavaScript I/O very similar to original ROOT code. All custom streamers are handled now in central place; a possibility to add custom streamers for user classes is provided. *JSROOT.TFile* class uses asynchronous HTTP requests [2] to retrieve data from ROOT files, therefore call-back functions are invoked for user notification when the requested object has been read from the file. Now I/O functionality can be used without graphic subsystem as shown on the figure 1.



```
{
  var filename = "https://root.cern.ch/js/files/hsimple.root";
  new JSROOT.TFile(filename, function(file) {
    file.ReadObject("hpxpy;1", function(obj) {
      console.log(“Object type name = “ + obj[‘_typename’]);
    });
  });
}
```

Figure 1. JavaScript code to an read object from the ROOT file

Also support for old ROOT v4 files format has been implemented. This is valuable to read ROOT files, produced by Geant4.

2.2. Graphics improvements

Graphics has been reorganized in a way that each supported object class has dedicated JavaScript painter class. It handles all drawings and provides possibility for online update of the object view. Different interactive features have been introduced. Most important are:

- via object context menu several actions can be performed – changing draw options, toggling linear/log scale, show/hide statistic box
- information in statistic box is updated when axis range selection is changed
- elements like title or statistic box can be moved and resized

A significant performance increase has been achieved for 1-D and 2-D histograms. MathJax.js [3] can now be used for LaTeX output.

2.3. Use of JSROOT on HTML page

The whole JavaScript ROOT code is organized in several modules such as:

io - reading of binary ROOT files
2d - main 2D graphic, based on d3.js
jq2d - advanced 2D graphic with jQuery.js and jQuery-ui.js
3d - 3D graphic for TH2/TH3 with three.js
gui - generic user interface

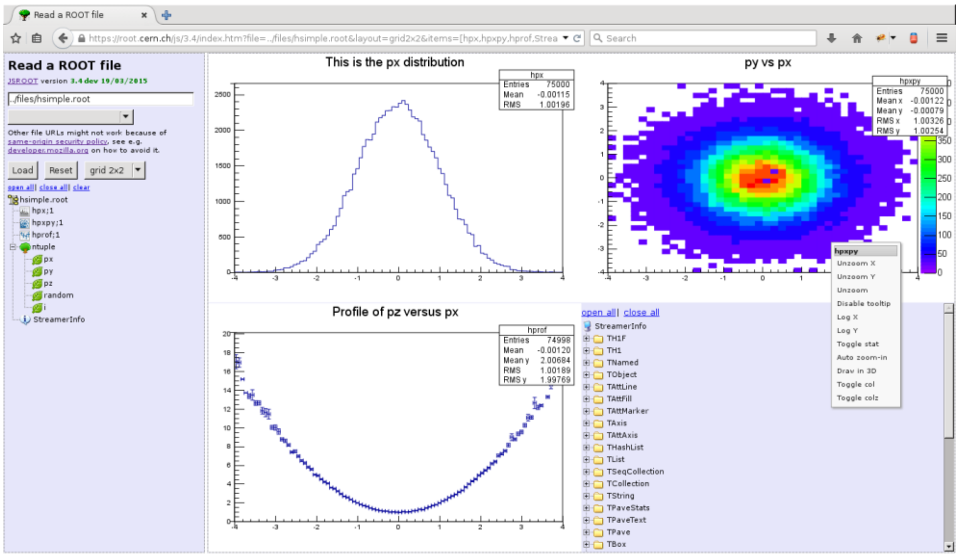
These modules can be specified when loading JSRootCore.js script on HTML page. The example Figure 2 shows complete HTML code for drawing an 1D histogram, retrieved from JSON file:

```
<!DOCTYPE html>
<html>
<head>
  <title>Reading object from the JSON file</title>
  <script src="https://root.cern.ch/js/3.4/scripts/JSRootCore.min.js?2d&onload=createGUI"
          type="text/javascript"></script>
  <script type="text/javascript">
    function createGUI() {
      var addr = "https://root.cern.ch/js/3.4/demo/hpx.json";
      var req = JSROOT.NewHttpRequest(addr, 'object', function(obj) {
         JSROOT.draw("drawing", obj, "hist");
      });
      req.send(null);
    }
  </script>
</head>
<body>
   <div id="drawing" style="width:800px; height:600px"></div>
</body>
</html>
```

Figure 2. HTML page to display histogram, read from JSON file

3. User interface

The original interface has been improved and extended. Several new layouts for objects drawing are provided. The JSROOT web site [4] can be directly used for viewing ROOT files from other web sites. Figure 3 show a JSROOT page with opened ROOT file and several displayed histograms.



https://root.cern.ch/js/3.4/index.htm?file=../files/hsimple.root&layout=grid2x2&items=[hpx,hpxpy,hprof,Strea

Read a ROOT file

JSROOT version 3.4 dev 19/03/2015

../files/hsimple.root

Other file URLs might not work because of same-origin security policy, see e.g. developer.mozilla.org on how to avoid it.

hpx	
Entries	75000
Mean	-0.00115
RMS	1.00196

hpxpy	
Entries	75000
Mean x	-0.00122
Mean y	-0.00079
RMS x	1.00326
RMS y	1.00254

hprof	
Entries	74998
Mean	-0.00120
Mean y	2.00684
RMS	1.00189
RMS y	1.99769

Figure 3. Browser with objects available from *hsimple.root* file. The objects hierarchy is on the left side, and several displayed histograms are on the right. Context menu for 2D histogram is activated

3.1. URL parameters

Different URL parameters can be specified when opening main JSROOT page:

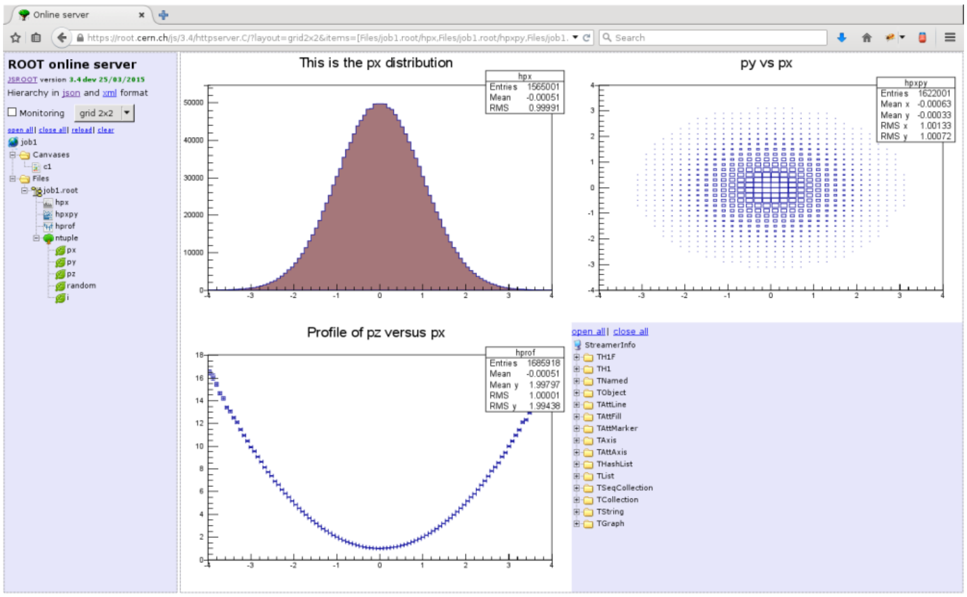
file(s)	– name of file(s) to open
item(s)	– item name(s) to display
opt(s)	– draw option for the item(s)
layout	– layout for drawings like grid or tabs
nobrowser	– do not display objects hierarchy
mathjax	– enable usage of MathJax.js
interactive	– enable/disable interactive features

Example of URL parameters usage:

https://root.cern.ch/js/3.5/file=../files/hsimple.root&item=hpx;1

4. Use with THttpServer class

JavaScript ROOT implements a generic user interface for *THttpServer* class as shown on figure 4. One can display and update objects from running ROOT applications; also command interface and remote TTree::Draw() are provided.



hpx	
Entries	1565001
Mean	-0.00051
RMS	0.99991

hpxpy	
Entries	1622001
Mean x	-0.00063
Mean y	-0.00033
RMS x	1.00133
RMS y	1.00072

hprof	
Entries	1685918
Mean	-0.00051
Mean y	1.99797
RMS	1.00001
RMS y	1.99438

Figure 4. Browser with objects available from running *tutorials/http/httpserver.C* macro. The objects hierarchy is on the left side, and several displayed histograms are on the right.

5. Conclusion

The JavaScript ROOT library allows reading and displaying ROOT objects in an efficient way, and its rendering is now very close to the original one provided by ROOT. Through its simple and flexible API JSROOT can be used in different web-based projects enabling ROOT-like graphics in the web browsers.

JavaScript ROOT is included in latest ROOT releases [5]. It also can be used directly online from project homepage [4]. Most recent developer code can be found on Github repository [6].

6. References

[1] B. Bellenot and S. Linev, "ROOT I/O in JavaScript", J. Phys.: Conf. Ser., Proceedings of CHEP2013, vol 513

[2] Asynchronous JavaScript and XML (Ajax), http://en.wikipedia.org/wiki/Ajax_(programming)

[3] MathJax homepage https://www.mathjax.org

[4] JavaScript ROOT homepage, https://root.cern.ch/js/

[5] ROOT homepage, https://root.cern.ch

[6] JavaScript ROOT developer repository, https://github.com/linev/jsroot