




ALICE feedback to the ROOT Program of Work



P. Hristov
14/06/2013

Outline

- ▶ “Short” time frame considered, no discussion on the Run3 upgrade needs
- ▶ ROOT 5: general feedback
- ▶ ROOT 5: specific feedback presented during the ROOT workshop in March
 - ▶ The ALICE DQM Software and ROOT, Barthelemy von Haller & Adriana Telesca
 - ▶ ALICE analysis experience with root, Andrei Gheata & Jan Fiete Grosse-Oetringhaus
- ▶ ROOT 6: first experience

ROOT – the foundation of ALICE SW

- ▶ **ALICE adopted ROOT as baseline since 1998**
 - ▶ Birth of AliRoot – the offline framework
 - ▶ Using from the beginning the most important ROOT features
 - ▶ persistency model, reflection, containers, ...
 - ▶ Contributing to ROOT development in many areas
 - ▶ Geometry, 3D graphics, event display, VMC
- ▶ **ALICE analysis users were "born" with ROOT**
 - ▶ ROOT is today a tool for their daily work and is expected to just work...

- ROOT provided key features
 - C++ database interface
 - Plugin mechanism
 - GUI
 - Macros
 - XML
- We could have done the same with a palette of tools, here we got it all at once
- Quick, kind and effective support
- Still, anything can always be improved...

ROOT 5: feedback on organization

- ▶ **Stability of the Root tags**
 - ▶ We experienced some problems with the latest Root tags that led to an extensive verification procedure with the AliRoot analysis tags built against the current Root version in production and the new tag
- ▶ **Better “synchronization”**
 - ▶ Root meetings are usually overlapping with the ALICE software meetings
 - ▶ “Sudden” transition to GIT => changes on the ALICE built servers
 - ▶ Savannah->JIRA: Savannah was redirecting to JIRA but JIRA was not ready...

ROOT 5: general support

- ▶ Support for the network file access
 - ▶ We are using xrootd, etc. and we want to be sure that the Root interfaces are always up-to-date
 - ▶ ROOT integration with xrootd v.4 and especially the handling of data buffering
- ▶ Support for AliEn. We have to be sure that the interface is working correctly
- ▶ Support for Proof. The stability of the CAF is essential for many analyses
- ▶ “Recalibration” of rootmarks: the scaling between the rootmarks and hepSPEC was lost last year (2012)

ROOT 5: specific observations and requests

Histograms & histogram graphics

- ▶ **Histograms hierarchy is wrong**
 - ▶ TH2, TH3 inherit from TH1, but not THn, why not a base class? One cannot write a general function applying to TH1 and THn without specializing it.
 - ▶ Compatibility issues between different histogram types
- ▶ **Axes are not easy to manipulate**
 - ▶ Responsibilities are unclear (histo, pad, axis)
 - ▶ Time-based axes should be far easier to use
- ▶ **Objects ownership, e.g. THStack (main histo is destroyed, not the others)**
- ▶ **Improvements in the errors treatment, store statistical & systematic errors**
 - ▶ Styles editable, handle drawing of multiple histograms
 - ▶ Allow editing drawing order
 - ▶ Allow scaling the statistical uncertainties (how the plot will look at different statistics)
 - ▶ Allow resetting errors or scaling with arbitrary function (different than the scaling of the histogram)

Histograms & histogram graphics

- ▶ Histogram “groups”
 - ▶ Uniform graphics style for the components
 - ▶ Apply changes to all with a single operation, even if plotted in different pads
 - ▶ Correct re-scaling for fonts in labels, legends, axes for “connected” histograms in adjacent pads.
- ▶ Now different scaling needed to preserve aspect ratio and one wrong mouse move can spoil one hour of work TPaveStats:
possibility to display only fit info, w/o any item of the stats
(now at least one has to be kept)
- ▶ Possibility to compare histograms with different binning
- ▶ Possibility to scale/divide/shift graphs, make graphs similar to histograms, points with labels, etc.

Histograms & histogram graphics

- ▶ There are still many nice features out there worth looking at and integrating in ROOT
 - ▶ e.g pgfplots, very easy to integrate with LaTeX
- ▶ Provide additional exchange histogram formats
 - ▶ Some people using features from other applications like: matlab/gnuplot/pgfplots/origin/etc)
- ▶ Add support for color scatter plots (i.e. combination of COLZ & SCAT), for e.g. dE/dx plots

Fitting

- ▶ Sometimes root doesn't find a simple fit
 - ▶ “I think it is a general remark that root should move to the best known fitting procedure which there is and not even grant the option anymore to use any obsolete ones”
- ▶ The old fit panel is really obsolete
- ▶ Integral/IntegralError for fits are rarely calculated properly and not even an error is given if the IntegralError is 0 although the errors on the function itself are large
- ▶ Include Crystal ball function in standard root fitting functions
 - ▶ Currently only included in roofit (?)

2D graphics & GUI

- ▶ **GUI: powerful but writing them is tough, error prone and pretty verbose**
- ▶ **Improvements for legends**
 - ▶ Changing the line and marker of a given entry
 - ▶ Allow blank lines for comments
 - ▶ Change the order with the mouse (reflected in the drawing order of histograms)
 - ▶ Pre-fill existing content in the editor
- ▶ **Add more shapes like TEllipse, TDiamond + possibility to rotate them**
- ▶ **Markers**
 - ▶ More marker styles, complete the set empty/filled, user defined
 - ▶ Version with black outline and colored content

2D graphics & GUI

- ▶ Add some templates for styles
- ▶ Possibility to change the margins of a canvas interactively
- ▶ Possibility to embed vectorized graphics (e.g. eps) in a canvas without losing the vectorization properties (e.g. ALICE logo) as a default operation handled by ROOT
- ▶ Possibility to undo most operations
 - ▶ I.e. recording the changes done by the last operations which are revertible
- ▶ TCanvas::SaveAs
 - ▶ Possibility to save the text content in a separate tex file so that the style can be adapted by TeX/LaTeX (a la gnuplot)
 - ▶ SaveAs/Load also for histogram graphics (a la copy/paste via the clipboard)

ACLiC

- ▶ Extend ACLiC functionality to handle multiple files, controlled by a meta-file information (like a simplified Makefile)
 - ▶ Typically a single hierarchical level is enough
 - ▶ This would allow automatic support for PAR-like libraries

PROOF

- ▶ How to better address scalability and stability issues in many-user environment?
 - ▶ All worked well when we had 24/7 operator and few users, not the case now.
- ▶ People are in need of a fast-response system with fast-track queues for their tests
 - ▶ ... which are not so simple anymore and may involve several input/output files

Miscellaneous

- ▶ **Sometimes obscure or inconsistent class interfaces**
 - ▶ E.g. `TMessage::EnableSchemaEvolution()` does actually nothing if not used in conjunction with `TSocket`
 - ▶ E.g. `Zoom` is a property of `THI` != `draw` option and `log scale` are properties of `TPad`
- ▶ **ROOT @ Web**
 - ▶ View and manipulate ROOT objects like in a `TCanvas/GUI`
- ▶ **Using ROOT as a simple library (no `TApplication`)**
 - ▶ We want our core files and stack traces

Miscellaneous

- ▶ **Support**

- ▶ Too often the answer is « Why don't you do it this way? » instead of fixing the issue

- ▶ **C++ standard or ROOT standard ?**

- ▶ STL, Boost, Collections and types

ROOT 6 – a challenge?

- ▶ ALICE was always a pioneer in testing at large scale the most recent ROOT releases, or even the trunk
 - ▶ In a production environment this becomes more and more hard to do...
 - ▶ ROOT 5 will still be used for some time
- ▶ We need the stability of ROOT 5, but also some of the new features that will be requested/implemented
 - ▶ Depending on the migration efforts, we would need a development branch with well controlled and tested releases still kept in ROOT 5 for a while

ROOT 6: first experience

- ▶ Compilation is possible after modification of 240 files

- ▶ Removing

```
#ifdef __CINT__
```

```
...
```

```
#else
```

```
# error Not for compilation
```

```
#endif
```

- ▶ Problems (may be we need additional options to rootcling)
- ▶ `#pragma link C++ defined_in` “header.h” doesn’t work => explicitly list all the classes from the file as pragmas
- ▶ no access to protected and private nested structures/classes for the building of dictionaries
- ▶ warnings about non-implemented pragmas, i.e. `#pragma link off` all methods
- ▶ data model evolution pragmas commented out, i.e. `#pragma read sourceClass="AliHLTTriggerMenuItem" version="[1-3]" targetClass="AliHLTTriggerMenuItem:..."`
- ▶ CINT-specific code commented out, i.e. `G__lasterror_lineno()`
- ▶ Change in `TH1::Intergal(...)` interface

ROOT 6: first experience

- ▶ The compilation takes longer than before
- ▶ It is possible to load macros
- ▶ It is possible to compile macros
- ▶ It is possible to exit from the AliRoot session
- ▶ AliRoot expectedly doesn't work

```
root [0] TFile * f = TFile::Open("I1000167920023.36.root")
```

```
Warning in <TClass::TClass>: no dictionary for class AliRawEventV2 is available
```

```
Warning in <TStreamerInfo::BuildCheck>:
```

```
The StreamerInfo from I1000167920023.36.root does not match existing one (AliRawEventV2:1)
```

```
The existing one has not been used yet and will be discarded.
```

```
Reading the file I1000167920023.36.root will work properly, however writing object of  
type AliRawEventV2 will not work properly. Most likely the version number  
of the class was not properly updated [See ClassDef(AliRawEventV2,1)].
```

```
Fatal: (cl==0 || info==0) && (cl!=0 || info!=0) violated at line 2118 of  
`/home/phristov/alice/root/io/io/src/TStreamerInfo.cxx'
```

```
aborting
```

- ▶ Work on the dictionaries is needed...

My ideal ROOT 7



Code name « redemption »

1. No major new features
2. Fix the (base) classes inheritance and interfaces
 - Hopefully it can even be backward compatible
3. Make ROOT web friendly
4. Make ROOT looks nice
 - GUI and graphics need a modern look and feel
 - Simplify GUI programming while you are at it

Prepare ROOT for the next 20 years and protect future users from mistakes made 20 years ago.