

Interpreter and Infrastructure Feedback

ROOT Workshop 2018
Sarajevo, Bosnia and
Herzegovina

ROOT
Data Analysis Framework
<https://root.cern>

- Cling interpreter
- Infrastructure
- Windows platform
- Teaching



- Cling and pyROOT appreciated for scripting capabilities
- Redeclare variable/function instead of crashing
 - <https://github.com/root-project/cling/issues/259>
- `root --nostrict`



ROOT installation

- No root access to machine: Conda, Spack, Docker
- Modularity
 - This is not new, CERN experiments has already requested multiple time such feature: a more modularized way to install ROOT or individual parts of ROOT
 - Decouple Cling from I/O, even ROOT without Cling?
 - Basically not to have so many dependencies from the system (?)
 - Functionality-wise, sometimes having more things is not beneficial
 - Production systems, security (“no compilers here”)



Minimal ROOT ?

- Difficult and cumbersome to build and install ROOT **when they only need a subset of it**
- Minimal ROOT: can we motivate this?
 - What is minimal root?
 - What depends on what?
 - Invest ~1 year of work? Is it that important?
 - Why are people so focused on getting minimal parts of ROOT?
 - What's the benefit of that?
 - (Answer) Kyle - Fermilab: That's the right question, but because of other people does it, maybe that's is what they are expecting so from ROOT.



ROOT build / dev install

- Concerns about CMake modernisation
 - Some planned features might not be (easily?) implemented with CMake
 - Are we willing to contribute to CMake main repository?
 - CMake responsive to bug report. Equally responsive for features requests?
- Building ROOT libraries without dictionary generation?
- Depending on / linking to ROOT
 - Request for clear guide (e.g. How to build a minimal stand-alone app)
 - Make find_package for libraries
- Frustration over building large software (build times?)



ROOT package manager

- Unclear what is scope: what does it do, **what not**, why not industry solution
 - “Lazy building”
 - Does not want to interfere with the Distribution Package manager
 - Provide flexibility to the developers building custom parts of ROOT, how to decouple different functionalities in modules, and apply lazy building of those



Development tools

- Clang-tidy linter to find bugs (plus bugprone constructs) / improve style
 - We have clang-tidy in our CI.
 - Do we interact with it enough? (Now it's always green.)
 - Should we go through components 1 at a time?
 - Extend with ROOT style rules?
- Integration with IDE



- General feature support issues
 - E.g. multi-process (fork) doesn't work
- Please also add XRootD
- Add DaviX



- People were really surprised with some of the forthcoming features
- **Main risk:**
 - Many independent projects popping up to implement individual parts of ROOT in python.
 - Are pythonizations a better approach?
- LHCb has been developing pythonizations since years
 - However they appreciate the new way to add them (with decorators), specially the lazy loading at runtime



- Ownership / memory management
 - Mentioned on previous feedback sessions
- Interoperability with numpy/ Python data science ecosystem
- ROOT builds for both Python2 and Python3
 - Simultaneous support for both versions



- More guidelines/tutorials:
 - Macros
 - General C++
 - Modern C++
 - Coding guidelines, software design tips
 - Introduction for good/effective ROOT practice. How to start.
 - Jupyter notebooks / interactive tutorials (as shown in the jupyter presentation)
- Mechanism allowing to go to the same class doc for different ROOT versions
- Advertise how to depend on root in your project.
- Clean up doc (updated things).



Should we invest more on teaching ROOT and C++ ?
Would be appreciated ?

- Stronger community to solve problems with ROOT (similar to Stackoverflow)
 - The goal is to cover the lack of ROOT experts in some areas (as the LHCb survey showed)
- How to lower the hurdle for newbies that just started and come with “*silly*”/*obvious* questions ?
- More learning resources
 - Procedures for problems already solved
 - Register more tutorials (more? Or improve the access to them?)
- Does ROOT provide tools to do *task X* in a better way?
 - Better understanding of the available methods (?)