To Compile or not to Compile

S. Hageboeck (CERN, EP-SFT) for the ROOT team

To Compile or not to Compile

- Did you ever use the "+" to compile a macro?
- .L something.cc++ ? I use it all the time!
- Great, good habit!
- Sure, anybody runs without?!
- So if a tutorial would not compile, because somebody couldn't be bothered to include the correct headers, would it be annoying? (I know the answer is yes ...)
- It would be time for a ticket 😄

→ Tutorials **must be compilable**!

(cc) BY

Less includes = easier?

- If the includes would disappear from the tutorials, because "they might look confusing to beginners"
  And as a compensation, the "+"-thing would figure out what to include automatically, would that be better/no change/worse?
- Hmm. Easier for users, ***IF*** you guarantee it works with every version ever. **Always**.
- It comes with a cost of understanding for the user, but then, including a header doesn't make the understanding *that* much better, maybe.

(cc) BY

Portability?

- But can you make it foolproof?
- Note that of you wanted to write an analysis framework, you would have to figure out the headers yourself by reading docs. There wouldn't be a single example with includes in the tutorials.
- Also, those auto-includes would only work with root headers. If users write their own headers, they still need to include those.
- Then no
- Oh, and important: It would only work with the next root version, and new tutorials/user macros would break if you handed them over to somebody with an older ROOT version.
- Hard no!

(cc) BY

No-goes

- Lol, you are just like me. So what are the no-goes again?
  - If new things don't work with older versions.
  - If it might not work if you tried to run in some other way than "+".
  - 'cos then users have no idea this feature exists, so can’t possibly know what is the problem.
  - Because if it’s not in the tutorial, people will not know it exists

(cc) BY

Summary

- It seems that compiling**+portability** is more popular than people thought
- One way out:

```
#include <inc/rf101_basics.h>
void rf101_basics() {
  ...
}
```

- It's still less portable (imagine somebody moves the file), but it would at least compile in older ROOTs.

(cc) BY