Subject: Re: compilation of macros
Posted by Bertram Kopf on Thu, 03 Dec 2009 13:48:34 GMT
View Forum Message <> Reply to Message

Hi Florian,

Quote:
I think it is good to look for potential problems, so it would also make sense to take a look at all the warnings which show up during compilation. They already show some of the problems.

The macros are meant for steering the execution of the compiled code and nothing else. If you want to compile them this is fine to find some errors which wer hidden up to now by the behaviour of rootcint.
But i don't understand why you want to get rid of the interpreter and invent a new one. as you said the requires alot of changes and i don't know who should do this if the people not even clean their own code after the did some changes or care about warnings in their part of code.

as you already wrote, the creation of binaries or the compilation of the macros would result in an additional diagnostic tool for the QA of the code.
The idea of introducing a new interpreter is as follows:
With ROOTCINT I don't see the possibility to interact with the framework in a sufficient way. Apart from the general steering of the applications, it would be helpful if the interpreter is also able to interact in a comfortable way with the tasks, the sub-tasks of the tasks and -if needed- also with the objects which are treated within the (sub)tasks. Examples are the setting of parameters or parameter sets for each module, the enabling/disabling or the cloning of modules etc at runtime. For such purposes I see some limitations by using ROOTCINT. Am I wrong?

Best regards,
Bertram.