

---

Subject: Re: Strategy discussion on Track objects  
Posted by [Johan Messchendorp](#) on Fri, 01 May 2009 14:01:11 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Dear all,

Let's try not to get too excited on these issues and let's try to avoid to revive old struggles and fights we had in the past. Important is that we on a very short term come with a scenario which is used and usable by us all. As Christian already said, we all have the same interest: a workable track reconstruction in our framework. So lets get back to the real issue, namely the point which Ralf stated earlier:

Quote:The second point is more complex and is regarded to the unification of the track objects used by the finders and fitters in general. A point which was made is that a track object should be able to provide a propagation functionality. This however conflicts with the policy to have data classes only storing their data, as far as I think.

I have to admit that partly this is a religious issue. So let me be nasty and refer to the computing model which states that one should try hard to minimize the number of internal dependencies. Furthermore, as a personal preference, I would like to avoid to be in a situation in which I need to load hundreds of libraries or find/include the two hundred corresponding header files in order to be able to look or perform a simple analysis on a root-file generated by a pandaroot. I, therefore, believe (!) one should try to separate as much as possible data classes from "functionality" issues, to keep it transparent at the end of the day. Ok, I have to admit, that I am also grown up with this tradition. So, if the majority of the tracking group has a different opinion, I wouldn't have any problem, though.

Kind wishes,

Johan.

---