Subject: Re: Closing SVN access on Wednesday 11 February 2009
Posted by Bertram Kopf on Tue, 17 Feb 2009 14:47:31 GMT
View Forum Message <> Reply to Message

Hi Ola and all others,

Aleksandra Wronska wrote on Tue, 17 February 2009 14:00
as for Mohammad's point 4, indeed, one of my classes inherits from one of the stt classes, namely SttRecoHit. I want to use exactly the same code, so I saw no point doing copy&paste. Some time ago I suggested to create a bit more general class in another directory, e.g. in recotasks, such, that both stt and dch can make use of it without being dependent on each other. As this has never been done, either dch package must depend on stt or I duplicate the code of SttRecoHit and WirepointHitPolicy. I also find both solutions unsatisfactory. Any suggestions?

I fully agree with you: it would be good to have a more general (abstract) class for this purpose. And this is not only related to the Stt and Dch. I think that also some other classes of the remaining dectoctor parts need to make use of exactly the same code or of just a few modifications. Therefore it is the best solution to define such a general class in the sense that the relevant detector specific classes inherit from this general object. Another advantage of such a structure is that everything is defined only at "one" place which makes it easier to modify and to maintain the code. In my point of view it would make sense to discuss such a design issue within the tracking core group.

In this context I have another remark to Mohammad's point 3. I took a look into the pndbase directory and I am surprised that this base directory contains only detector specific data classes. I my point of view a base directory should only contain "base classes", i.e. abstract classes or classes without (more or less ) any dependencies, which can be used everywhere. An example would be a general RecoHit class as suggested by Ola. The dectector specific data classes should be defined in directories which are at least one layer below.

Best regards,
Bertram.