
Subject: Re: PROOF in FairRoot

Posted by [Radoslaw Karabowicz](#) on Tue, 03 Apr 2012 10:28:52 GMT

[View Forum Message](#) <> [Reply to Message](#)

Dear Stefano,

Sorry for late answer.

The first problem is already solved in the current trunk. The problem was connected with your calling of `fRun->Run(0,nEvents)` in the macro, where `nEvents = 0`, which resulted in:

FairRunAna::RunOnProof(): The chain seems to have 100 entries.

FairRunAna::RunOnProof(): There are 1 files in the chain.

FairRunAna::RunOnProof(): Starting inChain->Process("FairAnaSelector","",0,0)

I have fixed it to take the whole chain if the user does not specify number of events to analyze.

The second problem: I guess the code crashes because of problems in MVD/SDS. The thing is I did not fix the classes. I will commit the changes I have implemented there in order to make it running. Sorry for that.

The last question:

"Trying to use different tasks I had several errors, which I have solved simply putting `//!` for the private objects of the tasks, which however should not be streamed.

But reading your comment on the site I have not well understood if this is safe or no. And if it is not, how to fix it properly?"

I also do it sometimes. There may be only one problem: assume you have some steering variable in your task, like `fVerbose`. You normally set it in the macro either with a setter or in a dedicated constructor. You would like to have this `fVerbose` also set in your tasks running on PROOF with the value you specify in the macro. But if you use `//!`, this value will not be streamed and the default value from the default constructor will be taken. Now `fVerbose` is not very crucial for running of the tasks but for example in my `PndBarrelTrackFinder` I have members to choose which detectors to use to search for tracks. I set them in macro and it is very important to pass this information to the tasks on the PROOF nodes. Because of that I cannot just do `//!` on these members.

yours
radek
