Subject: Persitancy in Tasks
Posted by Ralf Kliemt on Wed, 17 Feb 2010 08:16:20 GMT
View Forum Message <> Reply to Message

Hello everyone,

Stefano asked me to implement a SetPersistance() function to my Digi tasks and make them switch on/off the saving of my digis (thus reducing disk space). In principle this is no problem to do, but I realised that wrapper tasks like PndMvdDigiTask can pass this only in a difficult way.
Bu I have a solution.
I propose to extend FairTask by a boolean flag with a Set function. The set function then has to iterate through all subtasks, like it is done for the verbosity flag. All Tasks then can unse this flag in the persistancy flag of their TClonesArray's IO. E.g like:
Quote:
ioman->Register("MVDStripDigis", "MVD", fStripArray, fPersistance);

How are the opinions on that?

Kind regards, Ralf.

Subject: Re: Persitancy in Tasks
Posted by Tobias Stockmanns on Wed, 17 Feb 2010 08:24:04 GMT
View Forum Message <> Reply to Message

Hi Ralf,

please also add a GetPersistance() function to get the status of the flag.

Cheers,

Tobias

Subject: Re: Persitancy in Tasks
Posted by StefanoSpataro on Wed, 17 Feb 2010 08:33:36 GMT
View Forum Message <> Reply to Message

Hi,
I am not so sure if this can be good or not.
If one Task is filling more than one TCA, in this way you can switch both of them ON or OFF, but if you want to have only one ON... then you cannot do it.

This happens in PndLheTrackFinder::

```
//_________________________________________________________________
void PndLheTrackFinder::Register() {
  //---
  FairRootManager::
    Instance()->Register("LheCandidate",
                    "Lhe", fFoundTracks, fCanPersistence);

  FairRootManager::
    Instance()->Register("LheCMPoint",
                    "Lhe",fCMHits, fCMPersistence);

}
```

where I need to separate the persistence of the two objects. And by deafault one is ON and the other is OFF.

Subject: Re: Persitancy in Tasks
Posted by Ralf Kliemt on Wed, 17 Feb 2010 09:05:32 GMT
View Forum Message <> Reply to Message

Hello,

To me it is clear that there are parts of the data which we never throw away. These shall not be using that flag. If one needs a better distinction there can always be a second Persistance flag fot this individual task.

Another way would be to make the flag an integer, setting a level. But this requires a good definition and extra care when registering the arrays.

Ralf.

Subject: Re: Persitancy in Tasks
Posted by Tobias Stockmanns on Wed, 17 Feb 2010 09:34:58 GMT
View Forum Message <> Reply to Message

I think Stefano makes a good point. One wants to switch on persistancy individually for each data object. Why not use the fDetectorType (fDataType) to select in the task which objects to store?

Cheers,

Tobias

Subject: Re: Persitancy in Tasks
Posted by Mohammad Al-Turany on Wed, 17 Feb 2010 09:55:48 GMT
View Forum Message <> Reply to Message

Hi,

Why not to add such flags in the ctor of the tasks that need to do this? then you are free to add as much as you need. For the case of wrapper tasks, one implement the necessary set methods which will pass the flags to the ctor of the task lists. What do you think about this?

regards

Mohammad

Subject: Re: Persitancy in Tasks
Posted by Ralf Kliemt on Thu, 18 Feb 2010 11:44:10 GMT
View Forum Message <> Reply to Message

Hi,

I like to have the constructors of the tasks clean. This helps a lot with inner structural changes which then don't afflict the macros.
However if you don't want a persistancy flag in FairTask I'll work around that.

Greetings from Bonn.
Ralf.
