Subject: FinishTask() funktion for uninitialized Tasks -> Bug or feature?
Posted by Dominik Steinschaden on Wed, 18 Jan 2017 14:29:01 GMT
View Forum Message <> Reply to Message

Hello,

Currently I'm programming a few new Classes/ Tasks for PandaRoot, and realized the following behavior.

In my "Init()" function I implemented some checks and if they fail a "kERROR" is returned. (mainly if the needed input Branches are not found)
As expected for this "uninitialized" Tasks the "Task::Exec()" is not processes subsequently.
BUT the Task::FinishTask() function is still called, which leads to a segmentation violation in my case, although this is exactly what I want to avoid by doing this checks during the initialization.

Is this working as expected?

regards Dominik

Subject: Re: FinishTask() funktion for uninitialized Tasks -> Bug or feature?
Posted by Ralf Kliemt on Wed, 18 Jan 2017 14:35:46 GMT
View Forum Message <> Reply to Message

Hi Dominik,

Although I'm no expert I think this behavior is ok. You want to have the possibility to remove the traces of your Task even if the initialization fails. In principal you can re-initialize correctly after the change of the run.

There is a FairSoft part on the GSI forums, so the question should be asked there, too.

Cheers!
  Ralf

Subject: Re: FinishTask() funktion for uninitialized Tasks -> Bug or feature?
Posted by Dominik Steinschaden on Wed, 18 Jan 2017 14:42:16 GMT
View Forum Message <> Reply to Message

ok I see that there is maybe sometimes a point in having the finishTask called even in such a case, ok.

so thanks for the answer.

I will just rearrange my Init() function. My plan was to first do the checks and if everything goes fine then all needed variables are initialized. therefore if the checks fail, the variables are not initialized for the finishTask()

Just wanted to save resources . . .

Subject: Re: FinishTask() funktion for uninitialized Tasks -> Bug or feature?
Posted by Radoslaw Karabowicz on Wed, 18 Jan 2017 14:46:25 GMT
View Forum Message <> Reply to Message

Hi Dominik,

There is also a FairTask::Finish() method, that is only called if the task is initialized...

yours
radek

Subject: Re: FinishTask() funktion for uninitialized Tasks -> Bug or feature?
Posted by Radoslaw Karabowicz on Wed, 18 Jan 2017 15:05:01 GMT
View Forum Message <> Reply to Message

Or, to make it even better, just check in your FinishTask() for the IsActive() flag.

Subject: Re: FinishTask() funktion for uninitialized Tasks -> Bug or feature?
Posted by Tobias Stockmanns on Wed, 18 Jan 2017 15:27:53 GMT
View Forum Message <> Reply to Message

Hi Dominik,

I do not know your code but it would be very easy either to save the output of Init() in a variable and put the FinishTask() part in an if-statement or have your pointers initialized to 0 (as you always should do) and check for that in your FinishTask().

Cheers,

Tobias

Subject: Re: FinishTask() funktion for uninitialized Tasks -> Bug or feature?
Posted by StefanoSpataro on Wed, 18 Jan 2017 15:51:35 GMT
View Forum Message <> Reply to Message

Hi,
a good praxis (or maybe THE praxis) is to initialize the variables in the constructor, and not in the Init function:

PndStuff::PndStuff() : PndFather(), a(0), b(0), object() {}

In this way also the compiler helps you to find what is missing. Variables should be never left uninitialized. Once you have them initialized, in the Init you can set the proper values, or leave them as they are, and you won't have problems in the FinishTask.

Subject: Re: FinishTask() funktion for uninitialized Tasks -> Bug or feature?
Posted by Dominik Steinschaden on Thu, 19 Jan 2017 09:42:48 GMT
View Forum Message <> Reply to Message

Thanks for the help.

Just for interested persons: the variable causing the problems was a pointer to a TClonesArray which is used to write the data into the outputfile.

InitStatus PndSuperTask::Init(){
....

 FairRootManager* ioman = FairRootManager::Instance();

 fOutArray = ioman->Register(fOutBranchName.Data(), "FairTimeStamp", "EventDetermination", fPersistence);

.....

And If no Output is generated I want to avoid to register an output branch. I will just use an additional If statement in the FinishTask() as suggested.

Dominik

Subject: Re: FinishTask() funktion for uninitialized Tasks -> Bug or feature?
Posted by StefanoSpataro on Thu, 19 Jan 2017 11:24:50 GMT
View Forum Message <> Reply to Message

But in PndSupertask costructor did you put something like

PndSuperTask::PndSuperTask : xxx, xxx, xxx, fOutArray(), xxx, xxxx

?