

---

Subject: Task (or macro, I dont know) stops after one event  
Posted by [Marcel Tiemens](#) on Tue, 21 Feb 2017 15:42:49 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hi,

I created a new macro to run a new task. I based the macro on existing macros, which I know to work. The task tries to get some information, using FairLinks, on a set of objects that are loaded into a TClonesArray.

The TClonesArray is initiated like this:

```
fClusterArray = (TClonesArray*) ioman->GetObject("EmcClusterTemp");  
with ioman a pointer to the FairRootManager. The task has the standard layout of an Init(),  
Exec(), and a FinishTask(), which cout's some info and draw the histogram that was filled  
during the Exec stage. They are all defined as public virtual methods in the task's header file.
```

The macro looks like this, where clusFile contains the objects that need to be loaded into the TClonesArray, and simFile and digiFile contain the destination of the FairLinks:

```
FairRunAna *fRun= new FairRunAna();  
fRun->SetInputFile(clusFile);  
fRun->AddFriend(simFile);  
fRun->AddFriend(digiFile);  
fRun->SetOutputFile(outFile);  
  
fRun->SetUseFairLinks(kTRUE);  
  
// ----- Parameter database -----  
FairRuntimeDb* rtdb = fRun->GetRuntimeDb();  
  
FairParAsciiFileIo* parIo1 = new FairParAsciiFileIo();  
parIo1->open(parFile_Ascii.Data(),"in");  
rtdb->setFirstInput(parIo1);  
  
FairParRootFileIo* parInput1 = new FairParRootFileIo();  
parInput1->open(parFile);  
rtdb->setSecondInput(parInput1);  
  
// -----  
fRun->RunWithTimeStamps();  
  
PndEmcInspectClusters* inspectorTask = new PndEmcInspectClusters(iVerbose);  
fRun->AddTask(inspectorTask);  
  
// ----- Intialise and run -----  
  
gRandom->SetSeed();  
fRun->Init();  
fRun->Run();
```

When running this macro, everything seems to work, except that it exits after processing the first event. ROOT acts like it is supposed to do that, but it isn't. Trying to provide different ranges to the call fRun->Run() doesn't change anything. Does anyone know what could be causing this behaviour?

---

---

Subject: Re: Task (or macro, I dont know) stops after one event

Posted by [Stefano Spataro](#) on Tue, 21 Feb 2017 15:48:02 GMT

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

---

Without PndEmcInspectClusters task is difficult to understand what is going wrong. Where can one find it? The problem stays there, and not in the macro.

---

Subject: Re: Task (or macro, I dont know) stops after one event

Posted by [Marcel Tiemens](#) on Tue, 21 Feb 2017 15:50:38 GMT

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

---

All right, I've attached the source code of the task.

### File Attachments

---

1) [PndEmcInspectClusters.cxx](#), downloaded 141 times

2) [PndEmcInspectClusters.h](#), downloaded 162 times

---

Subject: Re: Task (or macro, I dont know) stops after one event

Posted by [Tobias Stockmanns](#) on Tue, 21 Feb 2017 16:02:24 GMT

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

---

Dear Marcel,

the problem is that you are running in time stamp mode by setting `fRun->RunWithTimeStamps()`. If you do that, you cannot receive the data in your `exec` method automatically but you have to add something like this:

```
if (FairRunAna::Instance() != 0 && FairRunAna::Instance()->IsTimeStamp()){
    <yourArray> = FairRootManager::Instance()->GetData(fInBranchName, fFunctor,
    <StopTime>);
}
```

The `fFunctor` is a `BinaryFunctor` defined in `FairTSBufferFunctional`. You have to decide in which data packages you want to have your data. Either in fixed time packages, then you choose the `StopTime` functor with an absolute time, or all data within a certain time gap, then you choose `TimeGap` functor with time between two events.

You can find an example e.g. in `PndMvdRiemannTrackFinderTask`.

I hope it helps.

Cheers,

Tobias

---

Subject: Re: Task (or macro, I dont know) stops after one event

Posted by [Marcel Tiemens](#) on Tue, 21 Feb 2017 16:07:12 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Thanks, that did the trick!

---