
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();

FairParAsciiFilelo* parlo1 = new FairParAsciiFilelo();
parlo1->open(parFile_Ascii.Data(),"in");
rtdb->setFirstInput(parlo1);

FairParRootFilelo* parInput1 = new FairParRootFilelo();
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?
