
Subject: Re: crash in reco macro (timebased simulation)

Posted by [Jens Sören Lange](#) on Mon, 12 Aug 2013 09:51:48 GMT

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

Hi Tobias, thanks a lot.

All the run_*_sttcombi_timebased.C macros run fine now,
including the run_tracking_sttcombi.C.

But then a problem happens, if I (in the next step) switch OFF the timebased part in the in the
run_digi_sttcombi_timebased.C
(for comparison of time-based and non-time-based)

```
PndMvdDigiTask* mvddigi = new PndMvdDigiTask();  
// mvddigi->RunTimeBased(); // <- HERE  
mvddigi->SetVerbose(iVerbose);  
fRun->AddTask(mvddigi);
```

then the run_reco_sttcombi.C runs o.k.,
but the run_tracking_sttcombi.C crashes with

```
I- FairGeane::FairGeane: Geane is Initialized  
-W- PndMvdRiemannTrackFinderTask::Init: No Branch Names given with  
AddHitBranch(TString branchName)! Standard BranchNames taken!  
[INFO ] Branch: MVDHitsPixel not found in Tree  
[INFO ] Branch: MVDHitsPixel not found in Tree  
-W- PndMvdRiemannTrackFinderTask::Init: No hitArray for BranchName MVDHitsPixel  
[INFO ] Branch: MVDHitsStrip not found in Tree  
[INFO ] Branch: MVDHitsStrip not found in Tree  
-W- PndMvdRiemannTrackFinderTask::Init: No hitArray for BranchName MVDHitsStrip  
-I- PndMvdRiemannTrackFinderTask: Initialisation successfull  
[INFO ] The number of entries in chain is 1
```

```
*** Break *** segmentation violation
```

```
[...]
```

```
=====  
There was a crash (kSigSegmentationViolation).
```

```
This is the entire stack trace of all threads:  
=====
```

The lines below might hint at the cause of the crash.
If they do not help you then please submit a bug report at
<http://root.cern.ch/bugs>. Please post the ENTIRE stack trace
from above as an attachment in addition to anything else
that might help us fixing this issue.

```
=====  
#10 0x00002ab505f52cab in PndMvdRiemannTrackFinderTask::FillHitArray (  
    this=0x1d730ba0)  
    at /home/soeren/backup/panda/pandaroot/apr13/mvd/MvdTracking/PndMvdRiemannT  
rackFinderTask.cxx:196
```

```
#11 0x00002ab505f5316e in PndMvdRiemannTrackFinderTask::Exec (this=0x1d730ba0,
    opt=0x2ab501690b20 "")
    at /home/soeren/backup/panda/pandaroot/apr13/mvd/MvdTracking/PndMvdRiemannT
rackFinderTask.cxx:122
#12 0x00002ab4f725e483 in TTask::ExecuteTasks (this=0x1d303360,
    option=0x2ab501690b20 "")
    at /home/soeren/backup/panda/fairsoft/sep12/tools/root/core/base/src/TTask. cxx:312
#13 0x00002ab4f725e712 in TTask::ExecuteTask (this=0x1d303360,
    option=0x2ab501690b20 "")
    at /home/soeren/backup/panda/fairsoft/sep12/tools/root/core/base/src/TTask. cxx:275
#14 0x00002ab5015af351 in FairRunAna::RunTSBuffers (this=0x1d3030f0)
    at /home/soeren/backup/panda/pandaroot/apr13/base/FairRunAna.cxx:656
#15 0x00002ab5015af998 in FairRunAna::Run (this=0x1d3030f0, Ev_start=0,
    Ev_end=0)
    at /home/soeren/backup/panda/pandaroot/apr13/base/FairRunAna.cxx:491
=====
```

This seems not to be a problem of the time-based digis, but a problem of the Riemann track finder - but only in the case of non-time-based digis (for non-time-based digis it runs fine).

Again (to make it clear), if I run the run_tracking_sttcombi.C when the mvddigi->RunTimeBased() was switched ON before, everything runs o.k.

any idea ?

thanks and best regards, Soeren