

---

Subject: Error during PndTpcElectronicsTask  
Posted by [Tobias Weber](#) on Mon, 20 Dec 2010 08:51:50 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hi all,

I encountered a problem during the digitization. When running my digi.C I get the following error:

```
Error: Symbol #include is not defined in current scope X3872_digi.C:147:
Error: Symbol exception is not defined in current scope X3872_digi.C:147:
Syntax Error: #include <exception> X3872_digi.C:147:
Error: Symbol G__exception is not defined in current scope X3872_digi.C:147:
Error: type G__exception not defined
FILE:/home/webert/Documents/Diplomarbeit/X3872_tpc_noFT/./X3872_digi.C LINE:147
(int)0
*** Interpreter error recovered ***
```

By comenting I found out that it arises because of the PndTpcElectronicsTask.  
I am using a fresh installation of PandaRoot(rev. 10456) and the ext. packages from january.

Best Regards,  
Tobias

---

---

Subject: Re: Error during PndTpcElectronicsTask  
Posted by [Stefano Spataro](#) on Mon, 20 Dec 2010 08:59:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

This problem is known since at least six months. It seems it goes away running the "trunk" version of the external packages, but this is only a fast fix.  
Somebody whould dig inside the code to understand what is really going wrong there.

---

---

Subject: Re: Error during PndTpcElectronicsTask  
Posted by [Tobias Weber](#) on Mon, 20 Dec 2010 13:44:44 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hi Stefano,

Thank you for your quick reply. I updated my external packages to the trunk version and compiled it. But the digitization is still chrashing at the same event.

---

---

Subject: Re: Error during PndTpcElectronicsTask  
Posted by [Stefano Spataro](#) on Mon, 20 Dec 2010 14:13:55 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

I think TPC people should investigate this seriously.  
This prevents from running a good amount of events with TPC, and the problem does not appear with STT code.

---

Subject: Re: Error during PndTpcElectronicsTask  
Posted by [Tobias Weber](#) on Mon, 20 Dec 2010 14:26:09 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hi,

I just want to say that Mathias is using the same operating system and PandaRoot version as I do and he is not able to reproduce the problem.  
Furthermore the problem does not occur on my laptop.

---

Subject: Re: Error during PndTpcElectronicsTask  
Posted by [Stefano Spataro](#) on Mon, 20 Dec 2010 16:39:23 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

The problem is connected to the memory usage of TPC digitization, probably.

Still, it has to be clarified if it comes from TPC data objects or from Links.

Once I have tried to remove the inheritance of the Tpc objects from FairMultiLinkedData, restoring them to TObject, and the digitization proceeded without crashes; but of course, by removing the links, the data objects are smaller in size then it is easier to not fill the memory. It is possible that the memory is messed by the fact that TPC data objects are storing pointers to the previous objects, instead of only the index number, and maybe this could also raise problems. But a test to take out this dependence was never done, and I am not going to do it considering it requires to change the tpc code structure.

Meanwhile, I have found the same crash in the reco part. To overcome it, I have commented out all the SetLink calls in the lhe code and in TrackData, and I was able to run reco also for 10k events.

Running the pid, again the same crash. Together with Lia I have cleaned a bit the code, but the crash is still persistent. I have commented out the SetLinks in the PidCorrelator, still the same crashes. I have tried to take out the inheritance of VAbsMicroCandidate from FairMultiLinkedData, and now it is still running.

For sure, there are problems with the FairLinks. But I am not so sure they are the cause of the TPC crash.

Subject: Re: Error during PndTpcElectronicsTask  
Posted by [Felix Boehmer](#) on Mon, 20 Dec 2010 16:55:49 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Dear Stefano,

The problem seen by Tobias Weber could be really anything - to me this just looks like CINT stumbled into some kind of uncontrolled behavior (maybe due to bad alloc).

Generally, the cross-reference to other TPC objects via pointers was necessary at the time they were introduced, since no such mechanism existed prior to Tobias' FairLink approach. It maybe is not pretty, but is no bit worse than keeping index lists to TClonesArray entries - also in terms of memory consumption!

If the crashes are really connected to this but only appear when the FairLinks are used, then it looks like handling of objects members with pointer type is not done correctly inside the FairLinks (considering also that the cross-reference using pointers inside the TPC classes has been around for wuite some time).

Can the people who experience these crashes please try to reproduce this problem with and without FairLinks, keeping an eye on memory consumption at the time of the crash (a simple "top" should suffice). Also it might help to compile the macros used for a more sensible crash stack.

Right now it is really hard to hunt down the problem, as I have never seen these crashes for myself, nor am I familiar what happens inside the FairLinks in full detail.

Cheers

Felix

---

---

Subject: Re: Error during PndTpcElectronicsTask  
Posted by [Stefano Spataro](#) on Mon, 20 Dec 2010 17:18:16 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Just few comments from my side:

Felix Boehmer wrote on Mon, 20 December 2010 17:55Dear Stefano,

The problem seen by Tobias Weber could be really anything - to me this just looks like CINT stumbled into some kind of uncontrolled behavior (maybe due to bad alloc).

If the problem is really "anything", then I am wondering why it is since six months that we have it and nobody was able to fix it (or most probably almost nobody has tried).

Quote:

Generally, the cross-reference to other TPC objects via pointers was necessary at the time they were introduced, since no such mechanism existed prior to Tobias' FairLink approach. It maybe is not pretty, but is no bit worse than keeping index lists to TClonesArray entries - also in terms of memory consumption!

If the crashes are really connected to this but only appear when the FairLinks are used, then it looks like handling of objects members with pointer type is not done correctly inside the FairLinks (considering also that the cross-reference using pointers inside the TPC classes has been around for quite some time).

The feeling is that links increase the amount of allocated memory in the object in a non-linear way, and maybe it fights against the TPC data structure (which is the only code using pointers; we had something in EMC but we have taken them out).

It is a matter of fact that now this kind of crash appears in TPC and not with other code, at least in digitization.

Quote:

Can the people who experience these crashes please try to reproduce this problem with and without FairLinks, keeping an eye on memory consumption at the time of the crash (a simple "top" should suffice). Also it might help to compile the macros used for a more sensible crash stack.

I have already spent enough time on this, giving all the details in the forum on how to reproduce the crash. I could not check memory consumption because it takes hours before the crash appears. Links cannot be taken out so easily, because of FairHits inheritance. In my case, removing them but from PndTpcCluster, the macro worked. But again, running the code which create PndTpcCluster, I had again the same problem.

Quote:

Right now it is really hard to hunt down the problem, as I have never seen these crashes for myself, nor am I familiar what happens inside the FairLinks in full detail.

Does it mean that you have tried to run 10k DPM events and the consequent digitization without crashes at all?

---

Subject: Re: Error during PndTpcElectronicsTask  
Posted by [Felix Boehmer](#) on Mon, 20 Dec 2010 18:17:18 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Dear Stefano,

let's try to be a little more constructive. The "Symbol\_\_\_G exception" hints towards an uncaught system signal, most likely a bad alloc. Memory load seems to be the most likely reason for these crashes.

In a nutshell: The only way to approach this problem is by scrutinizing some key questions, beginning with memory consumption. The people who have a setup configuration where this problem persistently appears could start by checking the time evolution of memory consumption (a couple of minutes and using simple tools like top should be fine) in different

scenarios:

FairLinks in the chain  
FairLinks commented out

In this way we can immediately learn if we have a memory leak (continuous growth) or rather one single "bad" event leading to catastrophic memory allocation, and how the presence of the FairLinks affects this - verifying or ruling out your idea that the TPC structure "fights" the FairLinks and that lead to non-linear growth of memory load. Per se, the presence of pointers as member variables shouldn't do ANY harm, since they are nothing more than one 32 (64) bit data block each.

I think this is the baseline homework that has to be done. Please try and let me know.

Cheers

Felix

---

Subject: Re: Error during PndTpcElectronicsTask  
Posted by [Stefano Spataro](#) on Mon, 20 Dec 2010 20:47:38 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Felix Boehmer wrote on Mon, 20 December 2010 19:17: Dear Stefano,

let's try to be a little more constructive. The "Symbol\_\_\_G exception" hints towards an uncaught system signal, most likely a bad alloc. Memory load seems to be the most likely reason for these crashes.

The error comes from FairRootmanager, when trying to read (ReadEvent) or filling (ForceFill) a Tree. It is the TTree command the one who gives the exception which is not caught. This means that the data saved in the tree, or which are going to be saved, are somehow corrupted. Discussing with Mohammad, a missing empty constructor/destructor or some not initialized data member could be the cause, but as I was not able to find such a case in a glance. Maybe there is something else.

But let me repeat the question: have you tried and succeed running 10k dpm events?  
As far as I know this problem appears in Torino, at GSI and also in Bonn.

---

Subject: Re: Error during PndTpcElectronicsTask  
Posted by [Felix Boehmer](#) on Tue, 21 Dec 2010 10:53:47 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Dear Stefano,

we have simulated many thousand DPM events just before the last meeting at GSI during

testing of the pattern recognition, although I cannot provide you with the exact number. I will build a new clean trunk and test it again.

Quote: The error comes from FairRootmanager, when trying to read (ReadEvent) or filling (ForceFill) a Tree. It is the TTree command the one who gives the exception which is not caught. This means that the data saved in the tree, or which are going to be saved, are somehow corrupted. Discussing with Mohammad, a missing empty constructor/destructor or some not initialized data member could be the cause, but I was not able to find such a case in a glance. Maybe there is something else.

Please be a little more exact about this, and elaborate why you suspect this. For me the behavior you describe is really only compatible with the assumption that we run into memory overload because we a) have rare events where very large numbers of objects would be created, or b) we have a permanent memory leak somewhere, most likely caused by a faulty destructor.

I have never seen this error on my system - maybe because I use a 64 bit machine, I don't know.

I'll investigate it and keep you updated.

---

Subject: Re: Error during PndTpcElectronicsTask  
Posted by [Stefano Spataro](#) on Tue, 21 Dec 2010 11:11:46 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Felix Boehmer wrote on Tue, 21 December 2010 11:53: Dear Stefano,

we have simulated many thousand DPM events just before the last meeting at GSI during testing of the pattern recognition, although I cannot provide you with the exact number. I will build a new clean trunk and test it again.

This would be nice.

Quote:

Please be a little more exact about this, and elaborate why you suspect this. For me the behavior you describe is really only compatible with the assumption that we run into memory overload because we a) have rare events where very large numbers of objects would be created, or b) we have a permanent memory leak somewhere, most likely caused by a faulty destructor.

I would opt for option b), considering that if you run exactly the messy event you do not get the error. Then I would think it is due to the integral of all the previous events -> memory slowly increasing and producing a mess somewhere.

Quote:

I have never seen this error on my system - maybe because I use a 64 bit machine, I don't

know.

I have investigated both 32bit and 64bit (i.e. lenny64) architectures, finding the same crash in both of them. I don't know which machines Ralf or Tobias were using.

---

Subject: Re: Error during PndTpcElectronicsTask  
Posted by [Felix Boehmer](#) on Tue, 21 Dec 2010 15:42:39 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hi again,

I started out with the simulations of many thousand BoxGen Events (each multiplicity = 5).  
My first observations are:

I don't seem to be able to reproduce the crash  
I see strange memory consumption behavior

Memory consumption is noteworthy:

Memory load *\*slowly\** grows step-wise, compatible with the fact that the TClonesArrays in memory will always have the size given by the largest event... saturating at roughly 500 MB  
Memory load on these "plateaus" is stable -> no big memory leaks  
There *\*are\** bad events that blow up memory load so much my system begins to swap  
The load stays high for many events, then falls back to a reasonable baseline

This is strange behavior indeed. The fact that the memory load *\*drops\** again after some time after the bad event proves that the memory consumption can not be caused by objects that live in the TClonesArrays, since that size would never decrease again. Also it can't be temporary events of one event, since they would have to disappear before the next event is processed, which is not what I see. The current guess is that the caching of the out-TTree is causing this...

I'll look into the TPC container classes as well as the FairLinks mechanism. There will probably be some redesigning. This is going to take a while.  
Meanwhile I would appreciate any input from other users that extends "there is something wrong with the TPC classes and the FairLinks"

Cheers

Felix

---

---

Subject: Re: Error during PndTpcElectronicsTask  
Posted by [Stefano Spataro](#) on Tue, 21 Dec 2010 17:23:01 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Felix Boehmer wrote on Tue, 21 December 2010 16:42: Hi again,

I started out with the simulations of many thousand BoxGen Events (each multiplicity = 5).

Could you please tell exactly which macros are you running? I want to test your sample. I have seen that for some reason it is easier to get the error with DPM than by using particle gun. Have you tried to run also dpm?  
And, are you using jan10 external packages or the trunk version?

Quote:

[\*] Memory load *\*slowly\** grows step-wise, compatible with the fact that the TClonesArrays in memory will always have the size given by the largest event... saturating at roughly 500 MB

If I have understood well, if we use XXXArray->Delete() (as we are doing in all the tasks now) the size of the TCA should restart from zero each event, and should not take the one from the largest event. I ask for confirmation (this is the reason why all the "Clear" were taken out).

Quote:

This is strange behavior indeed. The fact that the memory load *\*drops\** again after some time after the bad event proves that the memory consumption can not be caused by objects that live in the TClonesArrays, since that size would never decrease again.

I supposed this was connected to the Delete, but I am not so sure.

Quote:

Also it can't be temporary events of one event, since they would have to disappear before the next event is processed, which is not what I see. The current guess is that the caching of the out-TTtree is causing this...

Could be also, considering that the "faulty" part is connected to reading/writing the tree.

---

Subject: Re: Error during PndTpcElectronicsTask  
Posted by [Felix Boehmer](#) on Tue, 21 Dec 2010 18:01:55 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hi Stefano,

it's the standard runDigi macro from macro/tpc. I haven't tried with DPM yet. Can you please try and monitor the memory load with your macros?

Quote: If I have understood well, if we use XXXArray->Delete() (as we are doing in all the tasks now) the size of the TCA should restart from zero each event, and should not take the one



from the largest event. I ask for confirmation (this is the reason why all the "Clear" were taken out).

This is wrong. The difference between `clear()` and `delete()` is that with `clear()` the destructors of the objects in the `TClonesArray` are *\*not\** called (as opposed to `delete()`). However, the *\*allocated\** size of the `TClonesArray` is still the maximum that was ever needed, no matter how much is actually needed by the current "residents".

Cheers

Felix

---