

Hi,
yesterday I have updated my trunk version, and I have seen (feb11 or trunk external packages) that the macros have a crash exiting. If i run macro/run/sim_complete_tpc.C (or stt), the macro seems executed properly but just at the end I have a segmentation violation:

Toggle Spoiler

```
**** GTRIGI: IEVENT=   10 IDEVT=   10 Random Seeds =   4357       0
[INFO ] FairPrimaryGenerator: 4 primary tracks from vertex (0.000000, 0.000000, 0.000000 )
Event Time = 0.000000 (ns)
-I- PndStack: Filling MCTrack array...
-I- PndStack: Number of primaries      = 4
      Total number of particles = 2231
      Number of tracks in output = 791
-I- PndStack: Updating track indizes.....stack and 14 collections updated.
DIGI EXECUTION *****
RuntimeDb: write container PndEmcGeoPar
*** PndEmcGeoPar written to ROOT file version: 2
RuntimeDb: write container PndSensorNamePar
*** PndSensorNamePar written to ROOT file version: 2
RuntimeDb: write container PndEmcDigiPar
*** PndEmcDigiPar written to ROOT file version: 2
RuntimeDb: write container PndEmcDigiNonuniformityPar
RuntimeDb: write container FairBaseParSet
*** FairBaseParSet written to ROOT file version: 2
RuntimeDb: write container PndMultiFieldPar
*** PndMultiFieldPar written to ROOT file version: 2
RuntimeDb: write container PndTransPar
RuntimeDb: write container PndDipole1Par
RuntimeDb: write container PndDipole2Par
RuntimeDb: write container PndSolenoid1Par
RuntimeDb: write container PndSolenoid2Par
RuntimeDb: write container PndSolenoid3Par
RuntimeDb: write container PndSolenoid4Par
RuntimeDb: write container PndGeoPassivePar
*** PndGeoPassivePar written to ROOT file version: 2
RuntimeDb: write container PndTpcGeoPar
*** PndTpcGeoPar written to ROOT file version: 1
RuntimeDb: write container PndGeoTofPar
*** PndGeoTofPar written to ROOT file version: 2
RuntimeDb: write container PndGeoMdtPar
RealTime=44.980026 seconds, CpuTime=44.740000 seconds
Test passed
All ok
(int)0

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

=====

There was a crash (#7 0x00a0b2ac in SigHandler ()).
This is the entire stack trace of all threads:

=====

```
from /lib/ld-linux.so.2
#0 0x008007a2 in __dl_sysinfo_int80 () from /lib/ld-linux.so.2
#1 0x010770f3 in __waitpid_nocancel () from /lib/tls/libc.so.6
#2 0x010207b9 in do_system () from /lib/tls/libc.so.6
#3 0x007f398d in system () from /lib/tls/libpthread.so.0
#4 0x00a089b3 in TUnixSystem::Exec ()
    from /data1/panda_software/fairsoft_feb11/tools/root/lib/libCore.so.5.28
#5 0x00a0ec31 in TUnixSystem::StackTrace ()
    from /data1/panda_software/fairsoft_feb11/tools/root/lib/libCore.so.5.28
#6 0x00a0b216 in TUnixSystem::DispatchSignals ()
    from /data1/panda_software/fairsoft_feb11/tools/root/lib/libCore.so.5.28
#7 0x00a0b2ac in SigHandler ()
    from /data1/panda_software/fairsoft_feb11/tools/root/lib/libCore.so.5.28
#8 0x00a0a47d in sighandler ()
    from /data1/panda_software/fairsoft_feb11/tools/root/lib/libCore.so.5.28
#9 <signal handler called>
#10 0x00000039 in ?? ()
#11 0x03028b83 in ~odbgstream (this=0x3041b20)
    at /data0/spataro/feb11/pandaroot/tpc/dbgstream.cxx:135
#12 0x03029734 in __tcf_1 ()
    at /data0/spataro/feb11/pandaroot/tpc/dbgstream.cxx:39
#13 0x010165a7 in exit () from /lib/tls/libc.so.6
#14 0x00a08aaf in TUnixSystem::Exit ()
    from /data1/panda_software/fairsoft_feb11/tools/root/lib/libCore.so.5.28
#15 0x0092097d in TApplication::Terminate ()
    from /data1/panda_software/fairsoft_feb11/tools/root/lib/libCore.so.5.28
#16 0x006c63e9 in TRint::Terminate ()
    from /data1/panda_software/fairsoft_feb11/tools/root/lib/libRint.so.5.28
#17 0x009c1645 in TCint::ProcessLine ()
    from /data1/panda_software/fairsoft_feb11/tools/root/lib/libCore.so.5.28
#18 0x009c16e4 in TCint::ProcessLineSynch ()
    from /data1/panda_software/fairsoft_feb11/tools/root/lib/libCore.so.5.28
#19 0x00921e96 in TApplication::ExecuteFile ()
    from /data1/panda_software/fairsoft_feb11/tools/root/lib/libCore.so.5.28
#20 0x00922207 in TApplication::ProcessFile ()
    from /data1/panda_software/fairsoft_feb11/tools/root/lib/libCore.so.5.28
#21 0x0092009f in TApplication::ProcessLine ()
    from /data1/panda_software/fairsoft_feb11/tools/root/lib/libCore.so.5.28
#22 0x006c488d in TRint::Run ()
    from /data1/panda_software/fairsoft_feb11/tools/root/lib/libRint.so.5.28
#23 0x08048d56 in main ()
=====
```

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 0x00000039 in ?? ()
#11 0x03028b83 in ~odbgstream (this=0x3041b20)
    at /data0/spataro/feb11/pandaroot/tpc/dbgstream.cxx:135
#12 0x03029734 in __tcf_1 ()
    at /data0/spataro/feb11/pandaroot/tpc/dbgstream.cxx:39
#13 0x010165a7 in exit () from /lib/tls/libc.so.6
#14 0x00a08aaf in TUnixSystem::Exit ()
    from /data1/panda_software/fairsoft_feb11/tools/root/lib/libCore.so.5.28
#15 0x0092097d in TApplication::Terminate ()
    from /data1/panda_software/fairsoft_feb11/tools/root/lib/libCore.so.5.28
#16 0x006c63e9 in TRint::Terminate ()
    from /data1/panda_software/fairsoft_feb11/tools/root/lib/libRint.so.5.28
=====
```

Root >

From gdb output it seems it comes from the destruction of dbgstream, the command delete rdbuf(). In theory nothing has changed in this file, therefore I suppose the violation is coming from something else (crash noticed on an Ubuntu10.10 and on a SL4.7).

Another problem that I have seen is the inclusion of some eventdisplay code inside the tpc/TestBench package. We are not compiling EVE on grid machines -> tpc code crashes there.

An "If" would help to solve this problem, such as in the global CMakeLists.txt:

```
FIND_PATH(TEvePath NAMES TEveEventManager.h PATHS
  ${SIMPATH}/tools/root/include
  NO_DEFAULT_PATH
)

If(TEvePath)
#find_package(OpenGL)
#If (OPENGL_FOUND AND OPENGL_GLU_FOUND)
  Message("Found TEve, so the eventdisplay will be compiled.")
  add_subdirectory(eventdisplay)
  add_subdirectory(PndEventdisplay)
#EndIf (OPENGL_FOUND AND OPENGL_GLU_FOUND)
EndIf(TEvePath)
```

I think something similar should be added also on the tpc CMakeLists/LinkDef, so that tpc can work also in the grid.

Subject: Re: Problems with TPC: dbgstream and PndTpcClustVis

Posted by [Johannes Rauch](#) on Tue, 05 Apr 2011 13:25:20 GMT

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

Hi,

the second problem is my fault. Sorry.
I'm working on it right now.

Subject: Re: Problems with TPC: dbgstream and PndTpcClustVis

Posted by [Johannes Rauch](#) on Tue, 05 Apr 2011 15:50:59 GMT

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

Dear Stefano,

the problem should be fixed now. I made an own library for the class that needs EVE.
Could you please check if everything builds properly now without EVE?

Thanks!

Subject: Re: Problems with TPC: dbgstream and PndTpcClustVis

Posted by [Stefano Spataro](#) on Tue, 05 Apr 2011 16:12:34 GMT

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

Hi,

everything was compiling also with EVE, only GRID machines were crashing. For the next GRID update, I suppose we have to wait for next tuesday

Subject: Re: Problems with TPC: dbgstream and PndTpcClustVis

Posted by [Mohammad Al-Turany](#) on Thu, 07 Apr 2011 17:40:59 GMT

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

Hi Stefano,

About your first problem, it seems to originate from the CMakeList.txt in the TPC directory. For some reason that I do not know, the dbgstream.cxx is added as a source to all libraries that are created from this file. If you keep only one in the TPC_SRCS, then the crash disappear.

cheers,

Mohammad

Subject: Re: Problems with TPC: dbgstream and PndTpcClustVis
Posted by [Felix Boehmer](#) on Fri, 08 Apr 2011 12:34:33 GMT

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

I removed the stuff from libTpc for now, working on a final solution

Subject: Re: Problems with TPC: dbgstream and PndTpcClustVis
Posted by [Stefano Spataro](#) on Tue, 19 Apr 2011 08:23:32 GMT

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

Hi,
today I have checked the dashboard from grid machines, and I have seen that all of them are crashing because TPC package is linked to RGL, which is a graphical library not installed on GRID machines (they have no graphics at all).
Would it be possible to put an "IF", so that this part is linked/compiled only when GL are present? (like PndTpcClustVis case).
Thanks in advance

Subject: Re: Problems with TPC: dbgstream and PndTpcClustVis
Posted by [Stefano Spataro](#) on Tue, 26 Apr 2011 07:06:29 GMT

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

the crash is still there.

[42%] Building CXX object
mvd/CMakeFiles/MvdTrk.dir/MvdTracking/PndTpcClustPlusRTFTask.cxx.o

/usr/bin/ld: cannot find -IRGL

Subject: Re: Problems with TPC: dbgstream and PndTpcClustVis
Posted by [Johannes Rauch](#) on Tue, 26 Apr 2011 14:19:43 GMT

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

Sorry,

I missed to put an if statement in the CMakeList where the target link libraries are set. There the Root Libraries with RGL and EVE were included.
It should build now also without RGL.

regards,

Johannes
