

Dear colleagues,

Here is some trouble to simulate Pandaroot-MC again.
I'm using box generator and last trunk version.
For electron and muon the simulation is going very well, but when I try to do same with replacing pion or proton! Then the running of MC corrupt

Quote:

```
//This is OK!  
FairBoxGenerator* boxGen = new FairBoxGenerator(11, 1);
```

Quote:

```
// segmentaion fault  
FairBoxGenerator* boxGen = new FairBoxGenerator(211, 1);
```

Has someone made new things in TParticlePDG.h?

Here is the message from pandaroot!

Quote:

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

```
=====
There was a crash (kSigSegmentationViolation).
This is the entire stack trace of all threads:
=====
```

```
#0 0x00007f3bbb085b0e in waitpid () from /lib/libc.so.6
#1 0x00007f3bbb0231f9 in ?? () from /lib/libc.so.6
#2 0x00007f3bbd31171a in TUnixSystem::Exec (this=0x23807f0,
  shellcmd=0xca4d9a8 "/home/donghee/GSI/fairsoft_jan10/tools/root/etc/gdb-backtrace.sh
18592 1>&2")
  at core/unix/src/TUnixSystem.cxx:1978
#3 0x00007f3bbd312012 in TUnixSystem::StackTrace (this=0x23807f0) at
  core/unix/src/TUnixSystem.cxx:2188
#4 0x00007f3bbd30f864 in TUnixSystem::DispatchSignals (this=0x23807f0,
  sig=kSigSegmentationViolation)
  at core/unix/src/TUnixSystem.cxx:1106
#5 0x00007f3bbd30d539 in SigHandler (sig=kSigSegmentationViolation) at
  core/unix/src/TUnixSystem.cxx:350
#6 0x00007f3bbd31549e in sighandler (sig=11) at core/unix/src/TUnixSystem.cxx:3428
#7 <signal handler called>
```

```

#8 0x00007f3bb42a4300 in TParticlePDG::Mass (this=0x0) at include/TParticlePDG.h:72
#9 0x00007f3ba7149c74 in TGeant3::SetTrack (this=0x71598a0, done=1, parent=35,
pdg=1000020040,
    pmom=0x7f3ba74872a8, vpos=0x7f3ba7487e20, polar=0x7fff2725e6c0,
    tof=3.98251956e-08, mech=kPHadronic,
    ntr=
0x7fff2725e704, weight=1, is=0) at TGeant3/TGeant3.cxx:6367
#10 0x00007f3ba715166c in gustep_ () at TGeant3/TGeant3gu.cxx:808
#11 0x00007f3ba71331f6 in jumpt0_ (iadr=0x7f3ba7486cc8) at minicern/jumptn.c:74
#12 0x00007f3ba6fcabf0 in g3track () at gtrak/gtrack.F:296
#13 0x00007f3ba714f958 in gutrak_ () at TGeant3/TGeant3gu.cxx:558
#14 0x00007f3ba6fcbd3c in gtreveroot () at gtrak/gtreveroot.F:168
#15 0x00007f3ba714f97a in gutrev_ () at TGeant3/TGeant3gu.cxx:578
#16 0x00007f3ba71331f6 in jumpt0_ (iadr=0x7f3ba7486ce0) at minicern/jumptn.c:74
#17 0x00007f3ba6e4f3c5 in g3trig () at gbase/gtrig.F:62
#18 0x00007f3ba71430ff in TGeant3::Gtrig (this=0x71598a0) at TGeant3/TGeant3.cxx:3177
#19 0x00007f3ba7149b07 in TGeant3::ProcessEvent (this=0x71598a0) at
TGeant3/TGeant3.cxx:6300
#20 0x00007f3ba7149a13 in TGeant3::ProcessRun (this=0x71598a0, nevent=100) at
TGeant3/TGeant3.cxx:6270
#21 0x00007f3bb1efb93d in FairMCApplication::RunMC (this=0x3565630, nofEvents=100)
    at /home/donghee/GSI/pandaroot/base/FairMCApplication.cxx:238
#22 0x00007f3bb1f1761b in FairRunSim::Run (this=0x2fcb490, NEvents=100, NotUsed=0)
    at /home/donghee/GSI/pandaroot/base/FairRunSim.cxx:368
#23 0x00007f3bb1f5c8de in G__FairDict_681_0_5 (result7=0x7fff2726a880,
funcname=0x2fc78d0 "\001",
    libp=0x7fff2725fe70, hash=0) at /home/donghee/GSI/buildpanda/base/FairDict.cxx:10847
#24 0x00007f3bbc64b4e4 in Cint::G__ExceptionWrapper (funcp=0x7f3bb1f5c757
<G__FairDict_681_0_5>,
    result7=0x7fff2726a880, funcname=0x2fc78d0 "\001", libp=0x7fff2725fe70, hash=0)
    at cint/cint/src/Api.cxx:385
#25 0x00007f3bbc720b46 in G__execute_call (result7=0x7fff2726a880, libp=0x7fff2725fe70,
ifunc=0x2fc78d0,
    ifn=0) at cint/cint/src/newlink.cxx:2332
#26 0x00007f3bbc721479 in G__call_cppfunc (result7=0x7fff2726a880, libp=0x7fff2725fe70,
ifunc=0x2fc78d0,
    ifn=0) at cint/cint/src/newlink.cxx:2518
#27 0x00007f3bbc6faabc in G__interpret_func (result7=0x7fff2726a880,
funcname=0x23eee70 "Run",
    libp=0x7fff2725fe70, hash=309, p_ifunc=0x2fc78d0, funcmatch=1, memfunc_flag=1)
    at cint/cint/src/ifunc.cxx:5240
#28 0x00007f3bbc6d50bc in G__getfunction (item=0x23eda26 "Run(nEvents)",
known3=0x7fff2726e158,
    memfunc_flag=1) at cint/cint/src/func.cxx:2423
#29 0x00007f3bbc80a2e9 in G__getstructmem (store_var_type=112, varname=0x23ee650
"emcHitProd",
    membername=0x23eda26 "Run(nEvents)", tagname=0x23ee240 "fRun",
    known2=0x7fff2726e158,
    varglobal=0x7f3bbcae4540, objptr=2) at cint/cint/src/var.cxx:6591
#30 0x00007f3bbc7f9cf9 in G__getvariable (item=0x23eda20 "fRun->Run(nEvents)",
known=0x7fff2726e158,
    varglobal=0x7f3bbcae4540, varlocal=0x7fff27270f30) at cint/cint/src/var.cxx:5222

```

```

#31 0x00007f3bbc6c42a9 in G__getitem (item=0x23eda20 "fRun->Run(nEvents)") at
cint/cint/src/expr.cxx:1882
#32 0x00007f3bbc6c1b2f in G__getexpr (expression=0x2275a00 "fRun->Run(nEvents)")
    at cint/cint/src/expr.cxx:1464
#33 0x00007f3bbc752e4c in G__exec_function (statement=..., pc=0x7fff27270bcc,
piout=0x7fff27270bc8,
    plargestep=0x7fff27270bb8, presult=0x7fff27270a60) at cint/cint/src/parse.cxx:643
#34 0x00007f3bbc7633f3 in G__exec_statement (mparen=0x7fff27271484) at
cint/cint/src/parse.cxx:7083
#35 0x00007f3bbc6fe0fa in G__interpret_func (result7=0x7fff2727c440,
    funcname=0x2386860 "My_dvcs_mvd_run", libp=0x7fff27271a30, hash=1583,
p_ifunc=0x27ad480, funcmatch=1,
    memfunc_flag=0) at cint/cint/src/ifunc.cxx:6079
#36 0x00007f3bbc6d5dcf in G__getfunction (item=0x227fc60 "My_dvcs_mvd_run()",
known3=0x7fff2727c6d8,
    memfunc_flag=0) at cint/cint/src/func.cxx:2639
#37 0x00007f3bbc6c43c9 in G__getitem (item=0x227fc60 "My_dvcs_mvd_run()") at
cint/cint/src/expr.cxx:1894
#38 0x00007f3bbc6c1b2f in G__getexpr (expression=0x27cb960 "My_dvcs_mvd_run()")
    at cint/cint/src/expr.cxx:1464
#39 0x00007f3bbc6af9a8 in G__calc_internal (exprwithspace=0x2271710
"My_dvcs_mvd_run()")
    at cint/cint/src/expr.cxx:1066
#40 0x00007f3bbc76da2c in G__process_cmd (
    line=0x23d4598 ".X
/home/donghee/GSI/fairsoft_jan10/tools/root/etc/plugins/TVirtualStreamer
Info/P010_TStreamerInfo.C ", prompt=0x23866e8 "", more=0x23866e0, err=0x7fff272803a8,
rslt=0x7fff27280340)
    at cint/cint/src/pause.cxx:2277
#41 0x00007f3bbd2f7cf6 in TCint::ProcessLine (this=0x23866b0,
    line=0x23d4598 ".X
/home/donghee/GSI/fairsoft_jan10/tools/root/etc/plugins/TVirtualStreamer
Info/P010_TStreamerInfo.C ", error=0x7fff27281088) at core/meta/src/TCint.cxx:419
#42 0x00007f3bbd2f81f8 in TCint::ProcessLineSynch (this=0x23866b0,
    line=0x23d4598 ".X
/home/donghee/GSI/fairsoft_jan10/tools/root/etc/plugins/TVirtualStreamer
Info/P010_TStreamerInfo.C ", error=0x7fff27281088) at core/meta/src/TCint.cxx:486
#43 0x00007f3bbd1ef1ff in TApplication::ExecuteFile (file=0x7fff27281093
"My_dvcs_mvd_run.C",
    error=0x7fff27281088, keep=false) at core/base/src/TApplication.cxx:977
#44 0x00007f3bbd1ee910 in TApplication::ProcessFile (this=0x23caf60,
    file=0x7fff27281093 "My_dvcs_mvd_run.C", error=0x7fff27281088, keep=false)
    at core/base/src/TApplication.cxx:853
#45 0x00007f3bbd1ee82e in TApplication::ProcessLine (this=0x23caf60,
    line=0x7fff27281090 ".x My_dvcs_mvd_run.C", sync=false, err=0x7fff27281088)
    at core/base/src/TApplication.cxx:826
#46 0x00007f3bbb30549 in TRint::Run (this=0x23caf60, retrn=false) at
core/rint/src/TRint.cxx:407
#47 0x0000000000401234 in main (argc=1, argv=0x7fff27283238) at main/src/rmain.cxx:29
=====

```

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.

```
=====
#8 0x00007f3bb42a4300 in TParticlePDG::Mass (this=0x0) at include/TParticlePDG.h:72
#9 0x00007f3ba7149c74 in TGeant3::SetTrack (this=0x71598a0, done=1, parent=35,
pdg=1000020040,
    pmom=0x7f3ba74872a8, vpos=0x7f3ba7487e20, polar=0x7fff2725e6c0,
tof=3.98251956e-08, mech=kPHadronic,
    ntr=
0x7fff2725e704, weight=1, is=0) at TGeant3/TGeant3.cxx:6367
#10 0x00007f3ba715166c in gustep_ () at TGeant3/TGeant3gu.cxx:808
#11 0x00007f3ba71331f6 in jumpt0_ (iadr=0x7f3ba7486cc8) at minicern/jumptn.c:74
#12 0x00007f3ba6fcabf0 in g3track () at gtrak/gtrack.F:296
#13 0x00007f3ba714f958 in gutrak_ () at TGeant3/TGeant3gu.cxx:558
#14 0x00007f3ba6fcbd3c in gtreveroot () at gtrak/gtreveroot.F:168
#15 0x00007f3ba714f97a in gutrev_ () at TGeant3/TGeant3gu.cxx:578
#16 0x00007f3ba71331f6 in jumpt0_ (iadr=0x7f3ba7486ce0) at minicern/jumptn.c:74
#17 0x00007f3ba6e4f3c5 in g3trig () at gbase/gtrig.F:62
#18 0x00007f3ba71430ff in TGeant3::Gtrig (this=0x71598a0) at TGeant3/TGeant3.cxx:3177
#19 0x00007f3ba7149b07 in TGeant3::ProcessEvent (this=0x71598a0) at
TGeant3/TGeant3.cxx:6300
#20 0x00007f3ba7149a13 in TGeant3::ProcessRun (this=0x71598a0, nevent=100) at
TGeant3/TGeant3.cxx:6270
#21 0x00007f3bb1efb93d in FairMCApplication::RunMC (this=0x3565630, nofEvents=100)
    at /home/donghee/GSI/pandaroot/base/FairMCApplication.cxx:238
#22 0x00007f3bb1f1761b in FairRunSim::Run (this=0x2fcb490, NEvents=100, NotUsed=0)
    at /home/donghee/GSI/pandaroot/base/FairRunSim.cxx:368
=====
```

Subject: Re: Pandaroot crash due to TParticlePDG.h
Posted by [StefanoSpataro](#) on Sat, 24 Apr 2010 16:52:26 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,
this is connected to the fact that Geant3 is creating a particle, id 1000020040, I think it should
be an alpha, which is not stored inside TParticlePDG.
Mohammad some time ago wrote the solution (maybe to an Alicia message), but I don't
remember now how to fix it and where to find the original message.

However, maybe we should think about having our PndParticlePDG class, inheriting from
TParticlePDG, but with the ion stuff, in order to eliminate all these possible errors. I am not
sure if this can be done inside our code or if it requires changes in vmc code.

Subject: Re: Pandaroot crash due to TParticlePDG.h
Posted by [asanchez](#) on Sat, 24 Apr 2010 18:13:53 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi
the problem is not geant3 or geant4, because the ions created
as secondaries (parent=35, pdg=1000020040) are
created dynamically by the transport model itself.

So the problem must be somewhere else.

In case the particle is not defined geant3 send a message.
I will take a look into Geant3.cxx to check that the pdcCode is correct.

regards
ALicia.

Subject: Re: Pandaroot crash due to TParticlePDG.h
Posted by [asanchez](#) on Sat, 24 Apr 2010 18:21:49 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi again, concerning the ions staff,
If one want to generate ions at the level of the primary vertex
(as primaries), than the user must defined the particles
during the initialization(in the runMacro before calling the event generator task) by making use
of
FairParticle constructor.

So in principle an additional task for that it is not from my point of view necessary.
Only in the case that particles defined aby user are unstable, one should do
some modifications of Geant3.cxx where the alice people defined unstable particle
in a no t proper way.

cheers alicia.

Subject: Re: Pandaroot crash due to TParticlePDG.h
Posted by [asanchez](#) on Sat, 24 Apr 2010 18:58:56 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi
i was checking the code and
i found some inconsistency in the particle definition
between FairParticle and TGeant3.cxx.

In TGeant3.cxx alphas are defined with a pdgCode
(Line 7002)

determined by $\text{pdg} = 1000000000 + 10000 * z + 10 * a$;
which is the correct way.

Instead in FairParticle,

The constructor for standard ions

Line 129

`FairParticle(const char* name, Int_t z, Int_t a, Double_t mass, Int_t q, Bool_t stable, Double_t decaytime)`

uses $\text{fpdg} = 100000000 + 10000 * z + 10 * a$;

where the difference here is that 10000000 must be substituted by 10e9.

I'm not sure whether that is the problem, but at least the pdg codes for particles must be consistent everywhere within the code.

best regards

ALicia.

Subject: Re: Pandaroot crash due to TParticlePDG.h
Posted by [HosseinMoeini](#) on Wed, 26 Jan 2011 10:37:16 GMT
[View Forum Message](#) <> [Reply to Message](#)

Dear colleagues,

I am a new user of pandaroot starting to work on the EMC FwEndCap. I was wondering how I can make use of the FairBoxGenerator to generate ions and particles with undefined pdg codes like alpha, triton,.... I've already tried the following method for a few ions:
 $\text{pdg} = 1000000000 + 10000 * z + 10 * a$

But apparently it only works for alpha, triton, 3He and 4He. However, it doesn't work, for instance, for 179Au (using $\text{pdg} = 1000791970$)

Do you think if I'm using a not valid method to generate heavy ions and even light ones like alpha?
