

Hello,
after moving the charge in FairTrackPar from float to integer the genfit/geane code crashes in the following way:

Toggle Spoiler

```
*** Break *** floating point exception
(no debugging symbols found)
Using host libthread_db library "/lib/tls/libthread_db.so.1".
Attaching to program: /proc/15127/exe, process 15127
(no debugging symbols found)...done.
[Thread debugging using libthread_db enabled]
[New Thread -1208826176 (LWP 15127)]
(no debugging symbols found)...done.
(no debugging symbols found)...done.
(no debugging symbols found)...done.
(no debugging symbols found)...done.
(no debugging symbols found)...done.
(no debugging symbols found)...done.
(no debugging symbols found)...done.
(no debugging symbols found)...done.
0x0075f7a2 in _dl_sysinfo_int80 () from /lib/ld-linux.so.2
#1 0x00529f13 in __waitpid_nocancel () from /lib/tls/libc.so.6
#2 0x004d37b9 in do_system () from /lib/tls/libc.so.6
#3 0x0024198d in system () from /lib/tls/libpthread.so.0
#4 0x0096d4b7 in TUnixSystem::Exec () from
/home/spataro/july09/tools/root/lib/libCore.so.5.24
#5 0x0097314f in TUnixSystem::StackTrace () from
/home/spataro/july09/tools/root/lib/libCore.so.5.24
#6 0x0096fb4a in TUnixSystem::DispatchSignals () from
/home/spataro/july09/tools/root/lib/libCore.so.5.24
#7 0x0096fbd8 in SigHandler () from /home/spataro/july09/tools/root/lib/libCore.so.5.24
#8 0x0096ee55 in sighandler () from /home/spataro/july09/tools/root/lib/libCore.so.5.24
#9 <signal handler called>
#10 0x0190ea69 in FairGeaneUtil::FromSDToMars (this=0xbfe44d10, PC=0xbfe44c40,
RC=0xbfe44c60, H=0xbfe44d70, CH=0, SP1=1, DJ1=0xbfe452d8, DK1=0xbfe452f0,
PD=0xbfe44d50, RD=@0xbfe44b20) at
/home/spataro/july09/pandaroot/trackbase/FairGeaneUtil.cxx:1485
#11 0x018fc832 in FairTrackParP::FairTrackParP () at
/home/spataro/july09/tools/root/include/TVector3.h:280
#12 0x0631147b in GeaneTrackRep::extrapolate (this=0xca446f0, pl=@0xbfe464c0,
statePred=@0xbfe46640, covPred=@0xbfe46540)
at /home/spataro/july09/pandaroot/trackrep/GeaneTrackRep.cxx:152
#13 0x0302a5de in Kalman::processHit (this=0xbfe47ad0, tr=0xca43158, ihit=0, irep=0,
rejectOutlier=false)
at /home/spataro/july09/pandaroot/genfit/Kalman.cxx:248
#14 0x03029754 in Kalman::fittingPass (this=0xbfe47ad0, trk=0xca43158, direction=1) at
/home/spataro/july09/pandaroot/genfit/Kalman.cxx:140
#15 0x03028ef7 in Kalman::processTrack (this=0xbfe47ad0, trk=0xca43158) at
```

/home/spataro/july09/pandaroot/genfit/Kalman.cxx:38

The error disappears going back to an older svn release (6466). Then, maybe it could be better to restore the "floating" charge... or try to fix all the remaining stuff.

The problems appear within all the warnings after the "integer" change:

Toggle Spoiler

Scanning dependencies of target TrkBase

[79%] Building CXX object trackbase/CMakeFiles/TrkBase.dir/FairTrackPar.o

[79%] Building CXX object trackbase/CMakeFiles/TrkBase.dir/FairTrackParP.o

/home/spataro/july09/pandaroot/trackbase/FairTrackParP.cxx: In constructor

`FairTrackParP::FairTrackParP(Double_t, Double_t, Double_t, Double_t, Double_t, Double_t*, TVector3, TVector3, TVector3)':

/home/spataro/july09/pandaroot/trackbase/FairTrackParP.cxx:46: warning: converting to `Int\_t' from `Double_t'

/home/spataro/july09/pandaroot/trackbase/FairTrackParP.cxx: In constructor

`FairTrackParP::FairTrackParP(Double_t, Double_t, Double_t, Double_t, Double_t, Double_t*, TVector3, TVector3, TVector3, Double_t)':

/home/spataro/july09/pandaroot/trackbase/FairTrackParP.cxx:130: warning: converting to `Int\_t' from `Double_t'

/home/spataro/july09/pandaroot/trackbase/FairTrackParP.cxx: In member function `void FairTrackParP::SetTrackPar(Double_t, Double_t, Double_t, Double_t, Double_t, Double_t*, TVector3, TVector3, TVector3, TVector3, Double_t)':

/home/spataro/july09/pandaroot/trackbase/FairTrackParP.cxx:501: warning: converting to `Int\_t' from `Double_t'

[79%] Building CXX object trackbase/CMakeFiles/TrkBase.dir/FairTrackParH.o

/home/spataro/july09/pandaroot/trackbase/FairTrackParH.cxx: In constructor

`FairTrackParH::FairTrackParH(Double_t, Double_t, Double_t, Double_t, Double_t, Double_t, Double_t*)':

/home/spataro/july09/pandaroot/trackbase/FairTrackParH.cxx:44: warning: converting to `Int\_t' from `Double_t'

/home/spataro/july09/pandaroot/trackbase/FairTrackParH.cxx: In constructor

`FairTrackParH::FairTrackParH(FairTrackParP*, Int_t&)':

/home/spataro/july09/pandaroot/trackbase/FairTrackParH.cxx:188: warning: passing `Double\_t' for converting 7 of `void FairTrackParH::SetTrackPar(Double_t, Double_t, Double_t, Double_t, Double_t, Double_t, Int_t, Double_t*)'

/home/spataro/july09/pandaroot/trackbase/FairTrackParH.cxx: In member function `void FairTrackParH::SetTrackPar(Double_t, Double_t, Double_t, Double_t, Double_t, Double_t, Double_t*)':

/home/spataro/july09/pandaroot/trackbase/FairTrackParH.cxx:279: warning: converting to `Int\_t' from `Double_t'

[80%] Building CXX object trackbase/CMakeFiles/TrkBase.dir/FairGeaneUtil.o

[80%] Building CXX object trackbase/CMakeFiles/TrkBase.dir/TrackBaseDict.o

Linking CXX shared library ../lib/libTrkBase.so

[80%] Built target TrkBase

Scanning dependencies of target Geane

[80%] Building CXX object geane/CMakeFiles/Geane.dir/FairGeanePro.o

/home/spataro/july09/pandaroot/geane/FairGeanePro.cxx: In member function `Bool_t

FairGeanePro::Propagate(FairTrackParH*, FairTrackParH*, Int_t)':

/home/spataro/july09/pandaroot/geane/FairGeanePro.cxx:128: warning: passing `Float_t' for

```
converting 7 of `void FairTrackParH::SetTrackPar(Double_t, Double_t, Double_t, Double_t,
Double_t, Double_t, Int_t, Double_t*)'
/home/spataro/july09/pandaroot/geane/FairGeanePro.cxx: In member function `Bool_t
FairGeanePro::Propagate(FairTrackParP*, FairTrackParP*, Int_t)':
/home/spataro/july09/pandaroot/geane/FairGeanePro.cxx:244: warning: passing `Float_t' for
converting 7 of `void FairTrackParP::SetTrackPar(Double_t, Double_t, Double_t, Double_t,
Double_t, Double_t, Int_t, Double_t*, TVector3, TVector3, TVector3, TVector3)'
Linking CXX shared library ../lib/libGeane.so
[ 80%] Built target Geane
[ 80%] Generating trackrepDict.cxx, trackrepDict.h
Scanning dependencies of target trackrep
[ 80%] Building CXX object trackrep/CMakeFiles/trackrep.dir/GeaneTrackRep.o
/home/spataro/july09/pandaroot/trackrep/GeaneTrackRep.cxx: In constructor
`GeaneTrackRep::GeaneTrackRep(FairGeanePro*, const DetPlane&, const TVector3&, const
TVector3&, const TVector3&, double, int)':
/home/spataro/july09/pandaroot/trackrep/GeaneTrackRep.cxx:53: warning: passing `double' for
converting 5 of `FairTrackParP::FairTrackParP(TVector3, TVector3, TVector3, TVector3, Int_t,
TVector3, TVector3, TVector3)'
```

I don't have rights to write inside trackbase and I have no idea on how GeaneTrackRep (and relatives) is working, then I cannot fix it in the first way neither in the other.

I have reopened a ticket about this serious bug yesterday, but considering that there was no reply I think it has not worked properly, then I move the discussion in the forum.

Subject: Re: Problems with integer FairTrackPar charge
Posted by [Elwin Dijck](#) on Wed, 16 Sep 2009 18:49:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

I don't know how all these tracking classes work, but it looks like a division by zero, probably because of truncating a float that was almost but not exactly +/- 1.0 to zero, when converting it to an integer on the lines in FairTrackParX that give the warnings.

Perhaps rounding the charges when they are calculated on those lines instead of implicitly truncating them to integers would help, just a guess though.

Subject: Re: Problems with integer FairTrackPar charge
Posted by [Mohammad Al-Turany](#) on Wed, 16 Sep 2009 23:45:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

I found few places where it has to be changed from Double_t to Int_t in geane and trackbase. it is now replaced in SVN r6547. can you please check if it is still crashing. for me it seems to work!

regards

Mohammad

Subject: Re: Problems with integer FairTrackPar charge
Posted by [StefanoSpataro](#) on Thu, 17 Sep 2009 10:04:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,
thanks, but unfortunately the problem still persists:

```
*** Event # 1
===== PndLheHitsMaker =====
Total number of hits for tracking: 88
Total number of tracks in TPC: 1
    Good tracks in TPC: 1
Working with 88 hits
found 1 tracks
finder : Real Time = 0.01 seconds Cpu Time = 0.00 seconds
===== PndLheTrackFitter =====
Number of tracks for fitting 1
```

```
*** Break *** floating point exception
(no debugging symbols found)
Using host libthread_db library "/lib/tls/libthread_db.so.1".
Attaching to program: /proc/9557/exe, process 9557
(no debugging symbols found)...done.
[Thread debugging using libthread_db enabled]
[New Thread -1208490304 (LWP 9557)]
(no debugging symbols found)...done.
(no debugging symbols found)...done.
(no debugging symbols found)...done.
(no debugging symbols found)...done.
(no debugging symbols found)...done.
(no debugging symbols found)...done.
(no debugging symbols found)...done.
0x0075f7a2 in _dl_sysinfo_int80 () from /lib/ld-linux.so.2
#1 0x00475f13 in __waitpid_nocancel () from /lib/tls/libc.so.6
#2 0x0041f7b9 in do_system () from /lib/tls/libc.so.6
#3 0x003e498d in system () from /lib/tls/libpthread.so.0
#4 0x0096d4b7 in TUnixSystem::Exec () from
/home/spataro/july09/tools/root/lib/libCore.so.5.24
#5 0x0097314f in TUnixSystem::StackTrace () from
/home/spataro/july09/tools/root/lib/libCore.so.5.24
#6 0x0096fb4a in TUnixSystem::DispatchSignals () from
/home/spataro/july09/tools/root/lib/libCore.so.5.24
#7 0x0096fbd8 in SigHandler () from /home/spataro/july09/tools/root/lib/libCore.so.5.24
#8 0x0096ee55 in sighandler () from /home/spataro/july09/tools/root/lib/libCore.so.5.24
#9 <signal handler called>
```

```

#10 0x052499fb in FairGeaneUtil::FromSDToMars (this=0xbfe972e0, PC=0xbfe97210,
RC=0xbfe97230, H=0xbfe97340, CH=@0xbfe972d4, SP1=1, DJ1=0xbfe97898,
DK1=0xbfe978b0, PD=0xbfe97320, RD=@0xbfe970f0) at
/home/spataro/july09/pandaroot/trackbase/FairGeaneUtil.cxx:1485
#11 0x05237836 in FairTrackParP::FairTrackParP () at
/home/spataro/july09/tools/root/include/TVector3.h:280
#12 0x06abe016 in GeaneTrackRep::extrapolate (this=0xd41a090, pl=@0xbfe98ae0,
statePred=@0xbfe98c60, covPred=@0xbfe98b60)
at /home/spataro/july09/pandaroot/trackrep/GeaneTrackRep.cxx:151
#13 0x0472f5de in Kalman::processHit (this=0xbfe9a0f0, tr=0xd417200, ihit=0, irep=0,
rejectOutlier=false)
at /home/spataro/july09/pandaroot/genfit/Kalman.cxx:248
#14 0x0472e754 in Kalman::fittingPass (this=0xbfe9a0f0, trk=0xd417200, direction=1) at
/home/spataro/july09/pandaroot/genfit/Kalman.cxx:140
#15 0x0472def7 in Kalman::processTrack (this=0xbfe9a0f0, trk=0xd417200) at
/home/spataro/july09/pandaroot/genfit/Kalman.cxx:38
#16 0x0342c858 in PndLheKalmanTask::Exec (this=0xc2776f8, opt=0x21f40a8 "") at
/home/spataro/july09/pandaroot/lhetrack/PndLheKalmanTask.cxx:238
#17 0x008fab25 in TTask::ExecuteTasks () from
/home/spataro/july09/tools/root/lib/libCore.so.5.24
#18 0x008fa921 in TTask::ExecuteTask () from
/home/spataro/july09/tools/root/lib/libCore.so.5.24
#19 0x02196596 in FairRunAna::Run (this=0x950ec00, Ev_start=0, Ev_end=200) at
/home/spataro/july09/pandaroot/base/FairRunAna.cxx:248
#20 0x021cdfbc in G__FairDict_592_0_5 (result7=0xbfea8340, funcname=0x950c090 "\001",
libp=0xbfe9ccf0, hash=0)
at /home/spataro/july09/cbuild/base/FairDict.cxx:9025
#21 0x00e44d4b in Cint::G__ExceptionWrapper () from
/home/spataro/july09/tools/root/lib/libCint.so.5.24
#22 0x00ed9be4 in G__execute_call () from /home/spataro/july09/tools/root/lib/libCint.so.5.24
#23 0x00ed9ef6 in G__call_cppfunc () from /home/spataro/july09/tools/root/lib/libCint.so.5.24
#24 0x00ebabbf in G__interpret_func () from /home/spataro/july09/tools/root/lib/libCint.so.5.24
#25 0x00ea94f4 in G__getfunction () from /home/spataro/july09/tools/root/lib/libCint.so.5.24
#26 0x00f8d865 in G__getstructmem () from /home/spataro/july09/tools/root/lib/libCint.so.5.24
#27 0x00f8535b in G__getvariable () from /home/spataro/july09/tools/root/lib/libCint.so.5.24
#28 0x00e8d4e2 in G__getitem () from /home/spataro/july09/tools/root/lib/libCint.so.5.24
#29 0x00e90477 in G__getexpr () from /home/spataro/july09/tools/root/lib/libCint.so.5.24
#30 0x00f095dc in G__exec_statement () from
/home/spataro/july09/tools/root/lib/libCint.so.5.24
#31 0x00e7b668 in G__exec_tempfile_core () from
/home/spataro/july09/tools/root/lib/libCint.so.5.24
#32 0x00e7c99f in G__exec_tempfile () from /home/spataro/july09/tools/root/lib/libCint.so.5.24
#33 0x00f1a8ea in G__process_cmd () from /home/spataro/july09/tools/root/lib/libCint.so.5.24
#34 0x0095b4b3 in TCint::ProcessLine () from
/home/spataro/july09/tools/root/lib/libCore.so.5.24
#35 0x0095b634 in TCint::ProcessLineSynch () from
/home/spataro/july09/tools/root/lib/libCore.so.5.24
#36 0x00890cdb in TApplication::ExecuteFile () from
/home/spataro/july09/tools/root/lib/libCore.so.5.24
#37 0x0089106b in TApplication::ProcessFile () from
/home/spataro/july09/tools/root/lib/libCore.so.5.24
#38 0x0088eed3 in TApplication::ProcessLine () from

```

/home/spataro/july09/tools/root/lib/libCore.so.5.24

#39 0x002c0361 in TRint::Run () from /home/spataro/july09/tools/root/lib/libRint.so.5.24

#40 0x08048d5a in main ()

Subject: Re: Problems with integer FairTrackPar charge
Posted by [Elwin Dijck](#) on Thu, 17 Sep 2009 11:14:07 GMT

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

The rounding issue is not fixed. My guess is that the lines where the charge calculation is done in two steps give a zero charge sometimes (like FairTrackParH.cxx:43-44) due to rounding errors. Perhaps changing it to something like `fq = (int)TMath::Sign(1.0, fQp)` or doing it in one step would prevent the problem.

Subject: Re: Problems with integer FairTrackPar charge
Posted by [Ralf Kliemt](#) on Thu, 17 Sep 2009 16:32:34 GMT

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

Hi all,

I found the `TMath::Nint(Float_t)` function which might be exactly what we look for.

I wonder if it is default to track only charges of ± 1 . Is it really not possible to track e.g. alphas with Panda later from `pbar->Nucleus`?

Kind regards, Ralf.

Subject: Re: Problems with integer FairTrackPar charge
Posted by [asanchez](#) on Fri, 18 Sep 2009 07:33:11 GMT

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

Hi
i have been tracking particles ion-like,
by obtaining the charge by using `GetChargelon(Int_t ion)`
in my class `PndHypDPatternRecoTask`.

If the question concerns whether geane can track particles with charge magnitude bigger than 1 or not, that i don't know because i was using `LSLTrackRep`. But i think it must be any problem. Long time ago, i remmember there were some problems by trying get rid of `pdgcode` of ions, but maybe by giving the charge magnitude as I'm doing could solve the problem.

best regards
Alicia.

```
Int_t PndHypDPatternRecoTask::GetChargelon(Int_t ion)
261 {
```

```
262  Int_t A,Z,L;
263
264  if(ion>10000000000&&(ion<10100000000))
265  { ion -= 10000000000;
266    Z = ion/10000;
267    ion -= 10000*Z;
268    A = ion/10;
269    cout<<" ion charge "<<Z<<endl;
270
271    return Z;
272
273  }

287 }
```

Subject: Re: Problems with integer FairTrackPar charge
Posted by [Stefano Spataro](#) on Fri, 18 Sep 2009 08:20:54 GMT
[View Forum Message](#) <> [Reply to Message](#)

Excuse me,
the problem here is not how much the charge of a track is; the tracking code does no differences between charge 1 and 2 (apart if one refits a track with another particle hypothesis, not done presently in the code). Setting the value of the charge to the correct place is pid job, tracking should only set the sign.
In propagation the parameter which is used is q/p, then geane should be able to propagate particles with higher charge.

Here the problem is that I have asked some time ago to set the charge as integer instead of float. This was done by Mohammad, but somewhere in the code there is some part of the code that is doing bad things... while before the "change" everything was running fine. At least this is my feeling.

Subject: Re: Problems with integer FairTrackPar charge
Posted by [Lia Lavezzi](#) on Fri, 18 Sep 2009 11:20:43 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi all,
to answer your questions about GEANE: geane is able to track any particle (so also particle with charge different from +1 and -1 and neutral particles).

I will check with prof. Rotondi the interface to be sure that everything works fine: it surely works for charge +1 or -1,
it should be working also for other charged tracks (I made a quick test and it seems right) and it surely does not work for neutral particles (since we store as first track parameter q/p and in that case $q = 0$).

By the side of geane it should be enough to use, as Ralf said, the TMath::Nint(Float_t) function

if we want to have an integer charge (or to go back to double if we decide to).

Best regards,
Lia.

Subject: Re: Problems with integer FairTrackPar charge
Posted by [Mohammad Al-Turany](#) on Fri, 18 Sep 2009 21:47:45 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

just to get back to the original problem, namely the crash seen by Stefano after the change from Double_t to Int_t. I am not able to reproduce this on any of the following systems:

Mac Os X 10.5.8
Suse Enterprise 10.3
Ubuntu 9.04

all 64 bit systems, Stefano already told me privately that it crashes on the GSI cluster, but unfortunately today it was impossible to run on the GSI cluster! any way I will test this further and try to find out what is going on.

regards

Mohammad

Subject: Re: Problems with integer FairTrackPar charge
Posted by [StefanoSpataro](#) on Mon, 21 Sep 2009 09:43:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

Following the previous suggestions I have substituted, in FairTrackParP.cxx, all the:

```
fq = int(P * fQp)
```

with:

```
fq = (int)TMath::Sign(1.0, fQp)
```

Now the code does not crash anymore (with 200 events), then maybe the problem is solved. Maybe this fix should be done even on the helix parameters...

I do not have the rights to write inside trackbase, therefore somebody else should do these changes and commit the code...

Subject: Re: Problems with integer FairTrackPar charge
Posted by [Mohammad Al-Turany](#) on Mon, 21 Sep 2009 12:46:11 GMT

Hi,

I manage to reproduce the crash on Debian Etch 32, and it is in the FairGeaneUtil:

```
Quote:Program received signal SIGFPE, Arithmetic exception.
[Switching to Thread -1229228352 (LWP 18037)]
0xb3fe5bc1 in FairGeaneUtil::FromSDToMars (this=0xbfc1e124, PC=0xbfc1e0d8,
    RC=0xbfc1dfd0, H=0xbfc1e0f0, CH=0, SP1=1, DJ1=0xbfc1e6a0, DK1=0xbfc1e6b8,
    PD=0xbfc1e108, RD=@0xbfc1deb0)
    at /misc/turany/svn/pandaroot/trackbase/FairGeaneUtil.cxx:1485
1485    M65[0][0] = - SPU*PM2*PC[1]/(CH*PVW);
```

So as you can see it is a division by zero! and this comes from the CH = 0 which comes in this case from FairTrackParP:

```
fq= int (P * fQp);
```

As Stefano suggested replacing this with

```
fq = (int)TMath::Sign(1.0, fQp)
```

Solves the problem. I tried to print out the values for these two functions using 32 and 64 bit machines:

on 32-bit:

```
Quote:FairTrackParP::FairTrackParP fq = (int)TMath::Sign(1.0, fQp); P =2.046 fQp =
    -0.4887 fq = -1
FairTrackParP::FairTrackParP fq= int (P * fQp); P =2.046 fQp = -0.4887 fq = 0
```

and the same code on 64 bit:

```
Quote: FairTrackParP::FairTrackParP fq = (int)TMath::Sign(1.0, fQp); P =1.957 fQp = -0.5109
fq = -1
FairTrackParP::FairTrackParP fq= int (P * fQp); P =1.957 fQp = -0.5109 fq = -1
```

which explain why I could not reproduce this problem before!

Anyway, the change suggested by Stefano is now in SVN (-r 6568)

regards

Mohammad

Subject: Re: Problems with integer FairTrackPar charge
Posted by [Stefano Spataro](#) on Mon, 21 Sep 2009 13:19:45 GMT

It seems not it works.
I think that also the ticket can be closed.
Thanks

Subject: Re: Problems with integer FairTrackPar charge
Posted by [Felix Boehmer](#) on Mon, 21 Sep 2009 14:08:50 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

to me it looks like the int cast goes wrong. This would somehow explain why this is platform-dependant.

I replaced the corresponding fq assignments in FairTrackParP with statements like

```
if(fQp<0)
    fq = -1;
else
    fq = 1;
```

and it works now.

Subject: Re: Problems with integer FairTrackPar charge
Posted by [Stefano Spataro](#) on Tue, 27 Oct 2009 09:30:47 GMT
[View Forum Message](#) <> [Reply to Message](#)

Coming back to this topic,
I have seen that in GeaneTrackRep the charge is still defined as double, and then given to FairTrackParP where it is integer.
This produces the following warning:

```
/home/spataro/july09/pandaroot/GenfitTools/trackrep/GeaneTrackRep/GeaneTrackRep.cxx: In
constructor `GeaneTrackRep::GeaneTrackRep(FairGeanePro*, const GFDetPlane&, const
TVector3&, const TVector3&, const TVector3&, double, int)':
/home/spataro/july09/pandaroot/GenfitTools/trackrep/GeaneTrackRep/GeaneTrackRep.cxx:53:
warning: passing `double' for converting 5 of `FairTrackParP::FairTrackParP(TVector3,
TVector3, TVector3, TVector3, TVector3, TVector3, TVector3)'
```

Maybe also GeaneTrackRep should be fixed, in order to not produce problems.
In this case it seems this change should affect all the kalman tasks and also
PndSttPatternRecoTask/2.
Comments are welcome
