Subject: [FIXED] Bug in RhoCandList (?)
Posted by MartinJGaluska on Tue, 20 Aug 2013 10:49:10 GMT
View Forum Message <> Reply to Message

I wanted to write a function which is supposed to filter RhoCandList objects not by removing items out of a list, but by adding it to a new one. Therefore, I wrote the following function, but it does break when I try to run it and I really don't know why. Below the code is the error message.

```
int SelectGoodFitStatus(RhoCandList lOld, RhoCandList &lNew) {
 int removed = 0;
 for ( int ii=lOld.GetLength()-1; ii>=0; --ii ) {
  if ( !( lOld[ii]->GetRecoCandidate()->GetFitStatus()>0 ) ) {
   removed++;
  } else {
   lNew.Add( lOld[ii] );
  }
 }
 return removed;
}
```

Quote:
*** Break *** segmentation violation

==========
There was a crash.
This is the entire stack trace of all threads:
==========
#0 0x00007feeb86be44e in waitpid () from /lib/x86_64-linux-gnu/libc.so.6
#1 0x00007feeb864429e in ?? () from /lib/x86_64-linux-gnu/libc.so.6
#2 0x00007feeb957da77 in TUnixSystem::StackTrace() () from
/home/fairroot/apr13build/lib/root/libCore.so.5.34
#3 0x00007feeb9580353 in TUnixSystem::DispatchSignals(ESignals) () from
/home/fairroot/apr13build/lib/root/libCore.so.5.34
#4 <signal handler called>
#5 0x00007feeae41d740 in RhoCandList::Cleanup (this=0x904d8b0) at
/home/pandaroot_jun13/trunk/rho/RhoBase/RhoCandList.cxx:62
#6 0x00007feeae41d5f4 in RhoCandList::RhoCandList (this=0x904d8b0, l=...) at
/home/pandaroot_jun13/trunk/rho/RhoBase/RhoCandList.cxx:39
#7 0x00007feeae45795f in G__G__RhoDict_324_0_2 (result7=0x7fff14825b20,
funcname=0x184ecd0 "", libp=0x7fff1481b110, hash=0) at
/home/pandaroot_jun13/build/rho/G__RhoDict.cxx:8476
#8 0x00007feeb7ab0aab in Cint::G__ExceptionWrapper(int (*)(G__value*, char const*,
G__param*, int), G__value*, char*, G__param*, int) () from
/home/fairroot/apr13build/lib/root/libCint.so.5.34
#9 0x00007feeb7b563e1 in G__execute_call () from
/home/fairroot/apr13build/lib/root/libCint.so.5.34
#10 0x00007feeb7b567ce in G__call_cppfunc () from

/home/fairroot/apr13build/lib/root/libCint.so.5.34
#11 0x00007feeb7b37177 in G__interpret_func () from
/home/fairroot/apr13build/lib/root/libCint.so.5.34
#12 0x00007feeb7b23646 in G__getfunction () from
/home/fairroot/apr13build/lib/root/libCint.so.5.34
#13 0x00007feeb7af3df0 in G__define_var () from
/home/fairroot/apr13build/lib/root/libCint.so.5.34
#14 0x00007feeb7b8374e in G__exec_statement () from
/home/fairroot/apr13build/lib/root/libCint.so.5.34
#15 0x00007feeb7b34be5 in G__interpret_func () from
/home/fairroot/apr13build/lib/root/libCint.so.5.34
#16 0x00007feeb7b236ae in G__getfunction () from
/home/fairroot/apr13build/lib/root/libCint.so.5.34
#17 0x00007feeb7afd431 in G__getitem () from
/home/fairroot/apr13build/lib/root/libCint.so.5.34
#18 0x00007feeb7b033a4 in G__getexpr () from
/home/fairroot/apr13build/lib/root/libCint.so.5.34
#19 0x00007feeb7b86d23 in G__exec_statement () from
/home/fairroot/apr13build/lib/root/libCint.so.5.34
#20 0x00007feeb7b8ed82 in G__exec_loop(char const*, char*, std::list<G__FastAllocString,
std::allocator<G__FastAllocString> > const&) [clone .constprop.69] () from
/home/fairroot/apr13build/lib/root/libCint.so.5.34
#21 0x00007feeb7b87967 in G__exec_statement () from
/home/fairroot/apr13build/lib/root/libCint.so.5.34
#22 0x00007feeb7b352c7 in G__interpret_func () from
/home/fairroot/apr13build/lib/root/libCint.so.5.34
#23 0x00007feeb7b236ae in G__getfunction () from
/home/fairroot/apr13build/lib/root/libCint.so.5.34
#24 0x00007feeb7afd431 in G__getitem () from
/home/fairroot/apr13build/lib/root/libCint.so.5.34
#25 0x00007feeb7b033a4 in G__getexpr () from
/home/fairroot/apr13build/lib/root/libCint.so.5.34
#26 0x00007feeb7b0f0ac in G__calc_internal () from
/home/fairroot/apr13build/lib/root/libCint.so.5.34
#27 0x00007feeb7b967ef in G__process_cmd () from
/home/fairroot/apr13build/lib/root/libCint.so.5.34
#28 0x00007feeb9547bfa in TCint::ProcessLine(char const*, TInterpreter::EErrorCode*) () from
/home/fairroot/apr13build/lib/root/libCore.so.5.34
#29 0x00007feeb953e5e3 in TCint::ProcessLineSynch(char const*, TInterpreter::EErrorCode*)
() from /home/fairroot/apr13build/lib/root/libCore.so.5.34
#30 0x00007feeb94ad7b4 in TApplication::ExecuteFile(char const*, int*, bool) () from
/home/fairroot/apr13build/lib/root/libCore.so.5.34
#31 0x00007feeb94ac4ef in TApplication::ProcessLine(char const*, bool, int*) () from
/home/fairroot/apr13build/lib/root/libCore.so.5.34
#32 0x00007feeb910380a in TRint::Run(bool) () from
/home/fairroot/apr13build/lib/root/libRint.so.5.34
#33 0x0000000000400f6c 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.
==========
#5 0x00007feeae41d740 in RhoCandList::Cleanup (this=0x904d8b0) at
/home/pandaroot_jun13/trunk/rho/RhoBase/RhoCandList.cxx:62
#6 0x00007feeae41d5f4 in RhoCandList::RhoCandList (this=0x904d8b0, l=...) at
/home/pandaroot_jun13/trunk/rho/RhoBase/RhoCandList.cxx:39
==========

Root > Function SelectGoodFitStatus() busy flag cleared

Subject: Re: Bug in RhoCandList (?)
Posted by MartinJGaluska on Tue, 20 Aug 2013 10:50:44 GMT
View Forum Message <> Reply to Message

Ok, I found it. Root did not like that the first list was not passed as a reference. Is this intended behaviour or a bug?

Subject: Re: Bug in RhoCandList (?)
Posted by StefanoSpataro on Fri, 23 Aug 2013 15:14:57 GMT
View Forum Message <> Reply to Message

Good question...

Subject: Re: Bug in RhoCandList (?)
Posted by Klaus Götzen on Fri, 23 Aug 2013 15:39:19 GMT
View Forum Message <> Reply to Message

In principle it should work with the copy constructor as well I thought. I'm gonna take a look next week.

Subject: Re: [FIXED] Bug in RhoCandList (?)
Posted by Klaus Götzen on Mon, 26 Aug 2013 06:29:44 GMT
View Forum Message <> Reply to Message

Hi Martin,

I found a quite obvious (and presumably very old) bug in the copy constructor of RhoCandList which I just fixed in trunk.

Anyways, I think its faster to use reference parameters (which most likely is the reason, that

nobody was bugged by this bug up to now...)

Best,
Klaus

Subject: Re: [FIXED] Bug in RhoCandList (?)
Posted by MartinJGaluska on Mon, 26 Aug 2013 10:37:01 GMT
View Forum Message <> Reply to Message

Hi Klaus,

yes, you are right, I will probably continue passing it as a reference for efficiency reasons, but thank you very much for fixing the issue.

Martin
