
Subject: GENFIT objects in the ROOT Browser

Posted by [Sebastian Neubert](#) on Wed, 09 Apr 2008 16:02:44 GMT

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

Hi Dipak!

I just tried and in fact I can plot data from the ROOT Browser. For example if you click on cbmsim->Track->getMom()->Mag() this works beautifully.

Make sure that you have the dictionaries loaded when you do try this!

In general you can do much more powerfull things if you use draw commands from the command line.

See the plot macro for an example. Here you can even give parameters to the functions that you access in your draw command. Such functions are not even listed in the Browser (getCovElem(i,j) for example) because the Browser does not allow function parameters.

Cheers! Sebastian.

PS: It is amazing what you can do with such draw commands. In Compass I have written a class that manages complete decay chains and I use root draw commands to steer the calculation of all kinds of substates in the decay chain on the fly.

Subject: Re: GENFIT objects in the ROOT Browser

Posted by [Anonymous Poster](#) on Wed, 09 Apr 2008 16:29:44 GMT

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

Hi Sebastian,

It still doesn't work for me in TBrowser! I loaded the library using the your 'loadRecoLibs.C' then opened the browser. I can go upto:

cbmsim->Track->getMom().

As soon as I click on Mag() or px() or any variable it crashes giving the errors like: May be I'm doing something wrong.

ot [2]

*** Break *** segmentation violation

Using host libthread_db library "/lib/libthread_db.so.1".

Attaching to program: /proc/4369/exe, process 4369

Failed to read a valid object file image from memory.

done.

done.

[Thread debugging using libthread_db enabled]

[New Thread 16384 (LWP 4369)]

done.

done.

done.

done.

done.

done.

done.

```
done.
done.
0x40c90788 in waitpid () from /lib/libc.so.6
#1 0x40d188c0 in __DTOR_END__ () from /lib/libc.so.6
#2 0x40c29442 in do_system () from /lib/libc.so.6
#3 0x40ba3c5f in system () from /lib/libpthread.so.0
#4 0x401fa79f in TUnixSystem::Exec ()
    from /misc/cbmsoft/Debian3.1/mar08/fairsoft/tools/root/lib/libCore.so.5.18
#5 0x401fac63 in TUnixSystem::StackTrace ()
    from /misc/cbmsoft/Debian3.1/mar08/fairsoft/tools/root/lib/libCore.so.5.18
#6 0x401f85cc in TUnixSystem::DispatchSignals ()
    from /misc/cbmsoft/Debian3.1/mar08/fairsoft/tools/root/lib/libCore.so.5.18
#7 0x401f63a8 in SigHandler ()
    from /misc/cbmsoft/Debian3.1/mar08/fairsoft/tools/root/lib/libCore.so.5.18
#8 0x401fdefe in sighandler ()
    from /misc/cbmsoft/Debian3.1/mar08/fairsoft/tools/root/lib/libCore.so.5.18
#9 0x40ba2825 in __pthread_sighandler () from /lib/libpthread.so.0
#10 <signal handler called>
#11 0x40e140cc in TStreamerInfo::ReadBuffer<char**> ()
    from /misc/cbmsoft/Debian3.1/mar08/fairsoft/tools/root/lib/libRIO.so
#12 0x40dd46be in TBufferFile::ReadClassBuffer ()
    from /misc/cbmsoft/Debian3.1/mar08/fairsoft/tools/root/lib/libRIO.so
Root >
root [2] .qqqqqq
```

Subject: Re: GENFIT objects in the ROOT Browser
Posted by [Sebastian Neubert](#) on Thu, 10 Apr 2008 08:36:45 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi!

Ok. I can reproduce your error.

What I always did, was running the plotDemo.C and then directly afterwards the Browser. Then it works.

What also works (and I really suggest this as the best way to work) is to do the following:

```
sneubert@dimbula:/afs/e18/panda/SIM/sneubert/pandaroot$ root -l
root [0] .x recotasks/demo/loadRecoLibs.C
- RTDB container factory CbmBaseContFact
- RTDB container factory PndFieldContFact
- RTDB container factory PndPassiveContFact
- RTDB container factory PndMvdContFact
- RTDB container factory PndEmcContFact
- RTDB container factory PndDrcContFact
```

PSaid instance created... access via gSaid->f()

```
- RTDB container factory PndTpcContFact
root [1] TFile *_file0 = TFile::Open("demo.mcreco.root")
root [2] cbmsim->Draw("Track.getMom().Mag()")
<TCanvas::MakeDefCanvas>: created default TCanvas with name c1
root [3]
```

This behaviour of the Browser I have observed also in other contexts. It has to do with the way ROOT is reading in the objects from the file. Different things are happening behind the scenes when you access the tree directly or in the browser. The browser only reads part of the object into memory and when you have a bit more complicated functions it may not read all parts of the object that are necessary to perform the operation. The Tree seems to be more intelligent in that regard.

I think I once solved that in another case by setting the split level of the branch that I wanted to read to 1 or 2. So that you force ROOT to always load the complete object.

Cheers! Sebastian.