Subject: Usage of exceptions in CINT
Posted by Philipp Bielefeldt on Wed, 17 Feb 2016 11:41:54 GMT

Hi forum!

I am working on a CINT macro with FopiRoot, but I seem unable to get the exception handling done correctly. I set up a Kalman fitter kFitter and create a track that I want to process. As long as the fitter doesn't throw and exception, it seems to work -- however, in case the GFKalman fails, the following happens:

```
std::cout << "Entries in Track: " << NHitsInRecoTrack << std::endl;
try{
    kFitter->processTrack(recoTrack);
}
catch(const GFException& e){
    std::cerr << e.what() << std::endl;
    std::cout << "[WARN    ] Exception in track procession, next track" << std::endl;
    continue;
}
std::cout << "[INFO    ] Processed track!" << std::endl;
```

This is my code -- I put an cli output just before and after the try/catch-block.
The result (in a case an exception is thrown) is:

```
Entries in Track: 7
GFException thrown with excString:
RKTrackRep::RKutta ==> momentum too low: 1.08513 MeV
in line: 1215 in file:
/home/bielefeldt/g2fopiroot/GenfitTools/trackrep/RKTrackRep/RKTrackRep.cxx
with fatal flag 0
[INFO    ] Processed track!
```

Apparently, the exception is printed. But this also happens if I comment out the e.what-line; The "[WARN   ]..." and the "continue" seem to be ignored. However, even if I do not ask for the exception to be printed, it will be written to console.

The CINT documentation is a bit vague, but states
Quote:
Exception handling is implemented but with limitations. Object destruction may not be done properly in complex case. It is recommended not to use Exception Handling in interpreted code.

which is why I use interpreted code (loading with .L XYZ.C++).

I am quite puzzled -- maybe I am doing s.th. truly awful, but I do not understand how I am supposed to work properly with exception handling on the fitter. If anyone here can give me a little more understanding (or any useful reference), I'd be very thankful!

Cheers!
Philipp