
Subject: error pointer being freed was not allocated

Posted by [Tobias Stockmanns](#) on Tue, 15 Nov 2016 10:04:26 GMT

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

Dear all,

I am not sure what the reason for this error message is and whom to blame for it. With root6 I get often (not always) the following error at the end of the `sim_complete.C` macro (and all other following macros) of PandaRoot when I quit root with ".q".

```
root.exe(42661,0x7fffc36f93c0) malloc: *** error for object 0x7fe64dc46220: pointer being freed was not allocated
```

```
*** set a breakpoint in malloc_error_break to debug
```

When tracking this error with XCode it crashes in the destructor of TGeoManager in line 703 if (fShapes) {fShapes->Delete(); SafeDelete(fShapes);}

Going further down the delete is called in TCollection line 589.

Has anyone an idea how to fix this problem?

Cheers,

Tobias

Subject: Re: error pointer being freed was not allocated

Posted by [Florian Uhlig](#) on Tue, 15 Nov 2016 10:45:07 GMT

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

Hi Tobias,

somehow this is related to the geometry or maybe the way how we construct our geometries from different ROOT files.

We see the same also in CbmRoot but weren't able to find the reason for the strange behavior. Different people also including ROOT developers looked into the issue but couldn't find out what is the problem. By the way we get the error always.

For ROOT5 we created a patch to avoid the problem which is automatically applied when installing FairSoft. This is the reason why you never see the problem when using ROOT5.

A workaround which I found is to delete part of the geomanger before deleting the geomanager at the end of the macro. To simplify I create a function which I call at the end of the macros.

If you have an idea what could be the problem please let me know.

Ciao

Florian

```
/**
 * \ function RemoveGeoManager
 * There are some problems when deleting our geometries. In some cases
 * or combinations of geometries there is a double free of some memory
 * which results in a crash of ROOT. To avoid this we have patched one
 * ROOT class. With the newest ROOT6 version this isn't done any longer.
 * As a workaround to avoid the crash we delete two TObjArrays ourself
 * and then call the destructor of the TGeoManager at the end of the
 * macro. To simplify this one also can use this function.
 */
void RemoveGeoManager()
{
    if (gROOT->GetVersionInt() >= 60602) {
        gGeoManager->GetListOfVolumes()->Delete();
        gGeoManager->GetListOfShapes()->Delete();
        delete gGeoManager;
    }
}
```
