Subject: Question about using pointers to fill TClonesArray
Posted by MartinJGaluska on Thu, 18 Dec 2014 14:58:31 GMT
View Forum Message <> Reply to Message

Hello,

I am trying to find and fix memory leaks in my own code and have used code similar to the following line

PndTrackCand* myCand = new ((*fTrackCandArray)[i])PndTrackCand(trackFinder.GetTrackCand(i));

from   https://subversion.gsi.de/trac/fairroot/browser/pandaroot/trunk/mvd/MvdTracking/PndMvdRiemannTrackFinderTask.cxx line 162.

delete myCand;

is not present and I am wondering if it is necessary and if not, why.

Kind regards,
Martin

Subject: Re: Question about using pointers to fill TClonesArray
Posted by Klaus Götzen   on Thu, 18 Dec 2014 15:49:13 GMT
View Forum Message <> Reply to Message

Hi Martin,

as far as I understand the new at that point is a so-called 'new with placement', i.e. no new memory is allocated, but you write the new object at memory location, which already was use by an object of the same type (and size). This is much faster than new, and you don't need to delete (must not I guess!). This is the general clue of TClonesArrays.

Best,
Klaus

Subject: Re: Question about using pointers to fill TClonesArray
Posted by MartinJGaluska on Thu, 18 Dec 2014 15:53:38 GMT
View Forum Message <> Reply to Message

Thank you very much for the clarification!

Subject: Re: Question about using pointers to fill TClonesArray
Posted by Tobias Stockmanns on Thu, 18 Dec 2014 19:58:42 GMT

Dear Martin (and Klaus),

the new with placement creates a pointer at a position inside a TClonesArray. The owner of this object is the TClonesArray fTrackCandArray, myCand is a pointer pointing at the same memory position of the newly created PndTrackCand and is used to add some more information.

fTrackCandArray has to be deleted otherwise you will have in the next event still the PndTrackCands from the prvious event. This is done in the FinishEvent() method where the Delete() of the TClonesArrays is called.

I hope it gets a bit more clear but the TClonesArrays are a bit tricky to handle.

Cheers and a Merry Christmas,

Tobias

Subject: Re: Question about using pointers to fill TClonesArray
Posted by Klaus Götzen   on Thu, 18 Dec 2014 20:26:57 GMT

Hi Tobias,

thanks for pointing out the issue a bit more detailed!

Just like to give one additonal remark: If the objects stored in the TClonesArray don't allocate memory dynamically (like TString, std::vector, ...), you don't even need to call TCA->Delete(), which calls the destructor for all objects, but only TCA->Clear() which doesn't do that and thus is much faster.

However, PndTrackCand owns a std::vector, therefore Delete() has to be called I think.

Btw this is the reason, why the daughters (the only 'dynamic' quantity) of RhoCandidate are stored in an array with fixed size. Although being somehow unelegant, the destructor of RhoCandidate has never to be called by RhoFactory making it really fast. There actually was a severe memory leak before I got aware of that issue ...

Best and merry x-mas,
Klaus

Subject: Re: Question about using pointers to fill TClonesArray
Posted by Tobias Stockmanns on Thu, 18 Dec 2014 20:46:45 GMT

Dear Klaus,

the TClonesArray operates as you said. They allocate a block of memory which is dynamically extended if more storage is needed but never freed. If the TClonesArray is cleared the memory is internally marked as free. New data is then written into the same position as the previous data without the need to allocate new memory.

This only works if all objects in the TClonesArray have the same size which does not work for objects with container classes inside. For this purpose the TClonesArray uses a trick. It stores a pointer to the container classes while the data of the container classes are stored outside the memory block of the TClonesArray.

If you call a clear() for a TClonesArray with containers inside you get a memory leak only delete() calls the destructor of the container class which frees the memory of the container.

As I said, it is a bit tricky and it took me a while to figure it out.

Cheers,

Tobias

Subject: Re: Question about using pointers to fill TClonesArray
Posted by MartinJGaluska on Fri, 19 Dec 2014 12:42:24 GMT
View Forum Message <> Reply to Message

Thank you very much for the detailed explanation! It is indeed tricky to get the handling of the TClonesArrays right. The problem I was having with my code was connected with the storing of a class I implemented myself into a TClonesArray. In the end, I realized that I don't need to do that as I have other means of obtaining the output I need for parameter optimization and degugging and removed the corresponding code. This fixed the crashes I saw before.

Kind regards and happy holidays,
Martin