

---

Subject: [FIXED] Memory leak with TClonesArray and std::vector

Posted by [Klaus Götzen](#) on Fri, 23 Aug 2013 07:36:11 GMT

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

---

Hi,

this is more a ROOT related question, but maybe somebody from this club has an idea or knows about it.

The problem is that I observed quite a severe memory leak (just looking with 'top' how memory consumption of the job develops), apparently because the RhoCandidate stores the daughter pointers in a std::vector<RhoCandidate\*>.

I isolated the problem in a short ROOT macro looking like:

```
#include "TClonesArray.h"
#include <vector>
#include <iostream>

// very simple class to be used with a TClonesArray
class MyClass: public TObject
{
public:
    MyClass() {};
    ~MyClass() {};
    std::vector<int> fVec;
    ClassDef(MyClass,1)
};

void testtca(int num=100000)
{
    TClonesArray *tca = new TClonesArray("MyClass",1100);

    for (int i=0;i<num;++i)
    {
        if ((i%100)==0) cout <<i<<endl;
        tca->Clear();

        for (int j=0;j<1000;++j)
        {
            MyClass *c= new ((*tca)[j]) MyClass();
            c->fVec.push_back(42); // <- THIS IS THE BAD GUY!!1
            c->fVec.clear();
        }
    }
}
```

When you run this with 'root -l -b -q testtca.C+' you will see the memory growing very!! fast (be careful, you might want to reduce num).

When you comment out the line with 'push\_back(42)' it works perfectly well like you would expect it from a TClonesArray. The interesting part is, that even the 'fVec.clear()' directly after the push\_back does not help at all to avoid the problem (most likely because clear() does not really release the allocated memory of the vector). And I think even an appropriate cleanup in the destructor wouldn't really help, since the objects in a TCA are not deleted anyway, right?

I didn't find any way to fix this leak up to now. Does somebody here has an idea how to avoid that problem? Or a solution for a dynamic container for pointers different from std::vector?

Best,  
Klaus

---

---

Subject: Re: Memory leak with TClonesArray and std::vector  
Posted by [Florian Uhlig](#) on Fri, 23 Aug 2013 08:38:10 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hi Klaus,

you have to do a tca->Delete(); instead of a tca->Clear();

Clear will only release the memory used by the TClonesArray. Since the used vector is stored outside of the memory region used by the TClonesArray you delete only a pointer and you have a memory leak.

Delete will call the destructor of the object and such clear also memory used by the vector.

Ciao

Florian

---

---

Subject: Re: Memory leak with TClonesArray and std::vector  
Posted by [Klaus Götzen](#) on Fri, 23 Aug 2013 10:54:32 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Hi Florian,

thanks for the solution!

I thought I tested already with Delete() without success, but now it worked.

Maybe I have to think about a fixed array for the pointers to benefit from the TClonesArray speedup.

Best,  
Klaus

---