Subject: Re: Heap-leaks and polymorphic containers
Posted by mrjt on Tue, 13 May 2008 13:46:55 GMT
View Forum Message <> Reply to Message

luzr wrote on Mon, 12 May 2008 18:05Well, virtual destructor is absolutely required here.Hmm. I understand why this works (the implicit destructor isn't virtual):

```
struct Item : public Moveable<Item> {
 String name;

 virtual ~Item() { }
};

struct Collection : public Item
{
 Array<Item> items;
};

GUI_APP_MAIN
{
 Collection *col = new Collection();
 col->items.Add();
 Item *item = (Item *)col;
 delete item;
}
```

But what baffles me is why this works with Vector, but not Array (the Array's destructor doesn't get called but the Vector one does).

```
struct Item : public Moveable<Item> {
 String name;
};

struct Collection : public Item
{
 Vector<Item> items; // Heap leak if changed to Array
};

GUI_APP_MAIN
{
 Collection *col = new Collection();
 col->items.Add();
 Item *item = (Item *)col;
 delete item;
}
```