Subject: Why this one leaks ?
Posted by mdelfede on Sun, 11 Sep 2011 15:07:10 GMT
View Forum Message <> Reply to Message

```
#include <Core/Core.h>

using namespace Upp;

class Base : public Pte<Base>
{
 public:
  String BaseData;
  Base() { BaseData = "Sample data in Base class"; }
};

CONSOLE_APP_MAIN
{
 Ptr<Base> b = new Base;

}
```

Shouldn't be created element freed on Ptr destruction ?
Here the Upp log :

Heap leaks detected:

--memory-breakpoint__ 1120 : Memory at 0x7f70bdb87210, size 0x20 = 32
    +0 0x00007F70BDB87210 53 61 6D 70 6C 65 20 64 61 74 61 20 69 6E 20 42   Sample data
in B
    +16 0x00007F70BDB87220 61 73 65 20 63 6C 61 73 73 00 65 65 46 72 65 65   ase
class.eeFree

--memory-breakpoint__ 1119 : Memory at 0x7f70bdb872b0, size 0x28 = 40
    +0 0x00007F70BDB872B0 00 00 00 00 00 00 00 00 19 00 00 00 46 72 65 65   ....Free
    +16 0x00007F70BDB872C0 10 72 B8 BD 70 7F 00 00 10 72 B8 BD 70 7F 00 00

    +32 0x00007F70BDB872D0 19 00 00 00 00 00 1F 0F   ....

Subject: Re: Why this one leaks ?
Posted by mirek on Sun, 11 Sep 2011 15:22:06 GMT

mdelfede wrote on Sun, 11 September 2011 11:07

```
#include <Core/Core.h>

using namespace Upp;

class Base : public Pte<Base>
{
 public:
  String BaseData;
  Base() { BaseData = "Sample data in Base class"; }
};

CONSOLE_APP_MAIN
{
 Ptr<Base> b = new Base;


}
```

Shouldn't be created element freed on Ptr destruction ?
Here the Upp log :

You got it wrong way. b is set NULL on Base destructoin, not other way around.

U++ deliberately stays away from "smart pointers" used to manage shared resources.

Mirek

Subject: Re: Why this one leaks ?
Posted by mdelfede on Sun, 11 Sep 2011 15:27:52 GMT

Argh.... so I don't see the real usefulness of Ptr stuff, If I must remember to delete the pointed data....
Well, so no smart pointers in Upp,ok

Max

Subject: Re: Why this one leaks ?
Posted by mirek on Sun, 11 Sep 2011 18:17:15 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Sun, 11 September 2011 11:27Argh.... so I don't see the real usefulness of Ptr stuff, If I must remember to delete the pointed data....
Well, so no smart pointers in Upp,ok

Max

Well, Ptr is not intented to delete data, but to know that data was destructed...

Without it, GUI would be much harder to implement.

Mirek

Subject: Re: Why this one leaks ?
Posted by Mindtraveller on Mon, 12 Sep 2011 11:54:05 GMT
View Forum Message <> Reply to Message

You may use One<MyClass> container to have it auto-deleted.

Subject: Re: Why this one leaks ?
Posted by mdelfede on Mon, 12 Sep 2011 17:50:03 GMT
View Forum Message <> Reply to Message

Yep, I've understood that and I already used One.
The only caveat of One is that you must care when you pick it, but thinking a bit about it it's not a bad thing... makes me to code better and with more care

Max