Subject: This is heap leak?
Posted by fzx374cn on Fri, 29 Aug 2025 22:44:55 GMT
View Forum Message <> Reply to Message

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

using namespace Upp;
class Test : public TopWindow {
public:
 Test();
};

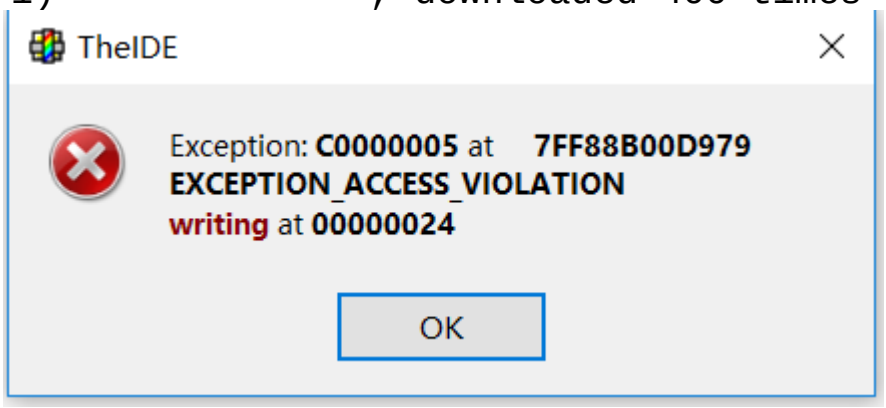
Test::Test()
{
 Ptr<Button> ptr;
 ptr = new Button;
}

GUI_APP_MAIN
{
 Test t;
 t.Run();
}
```

F5 run this code, when exit, popup a window,Is it heap leak?

File Attachments

1) , downloaded 400 times



Subject: Re: This is heap leak?
Posted by Oblivion on Tue, 02 Sep 2025 09:26:41 GMT
View Forum Message <> Reply to Message

Hi,

Yes that is a heap leak. You are creating a button on the heap but never deleting it.

Ptr<T> is used for pointing things, not for memory management.
If you want to create ctrls on the heap, you can use containers: One<Buttorn> or Array<Button>.
They can take care of the ownership and track the object life time.

Best regads,
Oblivion

Subject: Re: This is heap leak?
Posted by koldo on Tue, 02 Sep 2025 09:41:23 GMT
View Forum Message <> Reply to Message

Hi Fzx374cn

In addition Ptr goes with Pte.
This is a version of your example. It is not very meaningful but you can see that your pointer does not produce a heap problem, it just returns 0 if the pointed data (Foo data) is out of scope:

```
struct Foo : Pte<Foo> {
    Button but;
};

class Test : public TopWindow {
public:
 void Start() {
  Foo data;
  ptr = &data;
  str << "During: " << (void*)~ptr << "\n";
 }
 Ptr<Foo> ptr;
 String str;
};

GUI_APP_MAIN
{
 Test t;
 t.str << "Before: " << (void*)~t.ptr << "\n";
 t.Start();
 t.Run();
 t.str << "After:  " << (void*)~t.ptr;
 Exclamation(DeQtfLf(t.str));
}
```

This shows:
Before: 0x0
During: 0x6fad31f508
After: 0x0
