Subject: Heap Leaks Detected
Posted by MohamedMonem on Tue, 14 Apr 2009 23:27:59 GMT
View Forum Message <> Reply to Message

Hi All

can any one tell me whats make the application pop up this error (Heap Leaks Detected) after closing it !!

thanx in advance

Subject: Re: Heap Leaks Detected
Posted by cbpporter on Wed, 15 Apr 2009 07:09:51 GMT
View Forum Message <> Reply to Message

U++ keeps track of dynamic allocation and warns you if you have a memory leak. It means that you have not deleted something that you have newed. The warning will show up as long as you do not correct the memory leak. AFAIK there are some built-in means to help you detect where the leak is, but I have never used them.

By using U++ containers you can avoid having to delete you allocated data in more that 95% of cases (and in almost all of the above cases the call to new also), but U++ lacks an auto_ptr type for the rare cases when you do need to new a single item and delete it at some point.

Subject: Re: Heap Leaks Detected
Posted by mirek on Wed, 15 Apr 2009 07:33:01 GMT
View Forum Message <> Reply to Message

There is a bit of info:

http://www.ultimatepp.org/srcdoc$Core$Leaks$en-us.html

Mirek

Subject: Re: Heap Leaks Detected
Posted by mrjt on Wed, 15 Apr 2009 08:00:58 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Wed, 15 April 2009 08:09U++ keeps track of dynamic allocation and warns you if you have a memory leak. It means that you have not deleted something that you have newed. The warning will show up as long as you do not correct the memory leak. AFAIK there are some built-in means to help you detect where the leak is, but I have never used them.

By using U++ containers you can avoid having to delete you allocated data in more that 95% of cases (and in almost all of the above cases the call to new also), but U++ lacks an auto_ptr type for the rare cases when you do need to new a single item and delete it at some point.

Doesn't the One<> template provide this functionality?

Subject: Re: Heap Leaks Detected
Posted by TeCNoYoTTa on Wed, 15 Apr 2009 08:19:20 GMT
View Forum Message <> Reply to Message

thanks all ....but i think that the problem in our program is that there is a thread that don't close !!
how can we fix this

we made a thread that contains
while(true)
how can we change it to while the program is opened ?

thx in advance

Subject: Re: Heap Leaks Detected
Posted by mirek on Wed, 15 Apr 2009 09:40:41 GMT
View Forum Message <> Reply to Message

mrjt wrote on Wed, 15 April 2009 04:00cbpporter wrote on Wed, 15 April 2009 08:09U++ keeps track of dynamic allocation and warns you if you have a memory leak. It means that you have not deleted something that you have newed. The warning will show up as long as you do not correct the memory leak. AFAIK there are some built-in means to help you detect where the leak is, but I have never used them.

By using U++ containers you can avoid having to delete you allocated data in more that 95% of cases (and in almost all of the above cases the call to new also), but U++ lacks an auto_ptr type for the rare cases when you do need to new a single item and delete it at some point.

Doesn't the One<> template provide this functionality?

Definitely.

Although, according to U++ way of thinking (and as it progresses), One is considered to be a containter. And instead of 'new', I believe it is better to use 'Create' for in-place creation.

Mirek

Subject: Re: Heap Leaks Detected
Posted by cbpporter on Wed, 15 Apr 2009 11:07:08 GMT
View Forum Message <> Reply to Message

mrjt wrote on Wed, 15 April 2009 11:00
Doesn't the One<> template provide this functionality?
I stand corrected. Forgot about One. Just proves how rarely in need to keep track of resources through pointers .
luzr wrote on Wed, 15 April 2009 12:40
Definitely.

Although, according to U++ way of thinking (and as it progresses), One is considered to be a containter. And instead of 'new', I believe it is better to use 'Create' for in-place creation.

In place creation is really nice in general. Too bad C++ doesn't offer an in place creation mechanism, something like a delayed constructor.

Subject: Re: Heap Leaks Detected
Posted by mirek on Wed, 15 Apr 2009 14:11:09 GMT
View Forum Message <> Reply to Message

cbpporter wrote on Wed, 15 April 2009 07:07mrjt wrote on Wed, 15 April 2009 11:00
Doesn't the One<> template provide this functionality?
I stand corrected. Forgot about One. Just proves how rarely in need to keep track of resources through pointers .
luzr wrote on Wed, 15 April 2009 12:40
Definitely.

Although, according to U++ way of thinking (and as it progresses), One is considered to be a containter. And instead of 'new', I believe it is better to use 'Create' for in-place creation.

In place creation is really nice in general. Too bad C++ doesn't offer an in place creation mechanism, something like a delayed constructor.

As a matter of fact, I believe it does, the only problem is that for containers, the only constructor possible is default (paremeter-less) one.

But I guess that is only a samall price to pay. I generally consider parametric constructors either special case tools (e.g. Mutex::Lock) or suplementary (FileIn).

Mirek

Subject: Re: Heap Leaks Detected

Posted by sapiency on Sat, 06 Jun 2009 22:49:24 GMT
View Forum Message <> Reply to Message

TeCNoYoTTa wrote on Wed, 15 April 2009 10:19thanks all ....but i think that the problem in our program is that there is a thread that don't close !! how can we fix this

we made a thread that contains
while(true)
how can we change it to while the program is opened ?

thx in advance

just put

```
while (0 < Thread().GetCount())
{
  RLOG("[" << GetSysTime() << "." <<  "] "
    << __func__ << " Sleep(1) due of " << Thread().GetCount()
    << " Running threads"
    );
  Sleep(1);
}
```

in the destructor (for me it was the destructor of the class where the threads were created)
This will prevent to close the gui before the threads are closed.

regards

reinhard