Subject: Uninitialized class members.
Posted by Novo on Tue, 06 May 2008 15:31:35 GMT
View Forum Message <> Reply to Message

There are many uninitialized class members in U++. Valgrind complains about them a lot.

File Attachments
1) uninitialized_members.diff, downloaded 403 times

Subject: Re: Uninitialized class members.
Posted by mirek on Tue, 06 May 2008 17:44:44 GMT
View Forum Message <> Reply to Message

Not sure what valgrind really tests...

-ScrollBar::ScrollBar() {
+ScrollBar::ScrollBar() :
+style(NULL)
+{

"style" is initialized by SetStyle call in constructor.

-TreeCtrl::TreeCtrl()
+TreeCtrl::TreeCtrl() :
+ selectcount(0)

"selectcount" is initialized by Clear call in constructor.

Stopped there.... Finding them in .diff is too time consuming, maybe you could just post me the list of problems instead of .diff?

Mirek

P.S.: Do not get me wrong, valgrind is perhaps a good idea! But looks like it produces some false alarms too...

Subject: Re: Uninitialized class members.
Posted by Novo on Tue, 06 May 2008 18:34:08 GMT
View Forum Message <> Reply to Message

luzr wrote on Tue, 06 May 2008 13:44Not sure what valgrind really tests...

-ScrollBar::ScrollBar() {
+ScrollBar::ScrollBar() :
+style(NULL)
+{

"style" is initialized by SetStyle call in constructor.

Valgrind:

```
==30773==    at 0x5A10E3: Upp::ScrollBar::SetStyle(Upp::ScrollBar::Style const&)
==30773==    by 0x5DDB88: Upp::ScrollBar::ScrollBar() (in /export/home/ssikorsk/
==30773==    by 0x5F4699: Upp::ColumnList::ColumnList() (in /export/home/ssikors
==30773==    by 0x5F4A87: Upp::FileList::FileList() (in /export/home/ssikorsk/dv
==30773==    by 0x44A7D3: WorkspaceWork::WorkspaceWork() (in /export/home/ssikor
==30773==    by 0x45C1DB: Ide::Ide() (in /export/home/ssikorsk/dvlp/upp/svn/upp/
==30773==    by 0x482D8C: GuiMainFn_() (in /export/home/ssikorsk/dvlp/upp/svn/up
==30773==    by 0x484171: main (in /export/home/ssikorsk/dvlp/upp/svn/upp/out/GC
==30773==
```

Code:

```
ScrollBar& ScrollBar::SetStyle(const Style& s)
{
 if(style != &s) {
  style = &s;
  RefreshLayout();
  Refresh();
 }
 return *this;
}
```

That "if(style != &s)" looks very suspicious to me ...

Quote:
-TreeCtrl::TreeCtrl()
+TreeCtrl::TreeCtrl() :
+ selectcount(0)

"selectcount" is initialized by Clear call in constructor.

Valgrind:

```
==31285==    at 0x672D10: Upp::TreeCtrl::Dirty(int) (in /export/home/ssikorsk/dv
==31285==    by 0x67545A: Upp::TreeCtrl::Clear() (in /export/home/ssikorsk/dvlp/
==31285==    by 0x6756A8: Upp::TreeCtrl::TreeCtrl() (in /export/home/ssikorsk/dv
==31285==    by 0x6142BF: Upp::HelpWindow::HelpWindow() (in /export/home/ssikors
==31285==    by 0x42EE02: TopicCtrl::TopicCtrl() (in /export/home/ssikorsk/dvlp/
==31285==    by 0x45C91F: Ide::Ide() (in /export/home/ssikorsk/dvlp/upp/svn/upp/
==31285==    by 0x482D8C: GuiMainFn_() (in /export/home/ssikorsk/dvlp/upp/svn/up
==31285==    by 0x484171: main (in /export/home/ssikorsk/dvlp/upp/svn/upp/out/GC
==31285==
```

Code:

```
void   TreeCtrl::Clear()
{
 item.Clear();
 item.Add();
 item[0].linei = -1;
 item[0].parent = -1;
 item[0].canopen = true;
 freelist = -1;
 Dirty();
 cursor = anchor = -1;
 selectcount = 0;
}
```

Dirty() is called before initialization of select count.

Quote:
Stopped there.... Finding them in .diff is too time consuming, maybe you could just post me the list of problems instead of .diff?

I can post either diff or results of valgrind processing, which you can produce yourself ...

Quote:
P.S.: Do not get me wrong, valgrind is perhaps a good idea! But looks like it produces some false alarms too...

As you can see from examples above, valgrind is not giving false alarms.

Subject: Re: Uninitialized class members.
Posted by mirek on Tue, 06 May 2008 19:27:24 GMT
View Forum Message <> Reply to Message

Novo wrote on Tue, 06 May 2008 14:34luzr wrote on Tue, 06 May 2008 13:44Not sure what valgrind really tests...

-ScrollBar::ScrollBar() {
+ScrollBar::ScrollBar() :
+style(NULL)
+{

"style" is initialized by SetStyle call in constructor.

Valgrind:

```
==30773==    at 0x5A10E3: Upp::ScrollBar::SetStyle(Upp::ScrollBar::Style const&)
==30773==    by 0x5DDB88: Upp::ScrollBar::ScrollBar() (in /export/home/ssikorsk/
==30773==    by 0x5F4699: Upp::ColumnList::ColumnList() (in /export/home/ssikors
==30773==    by 0x5F4A87: Upp::FileList::FileList() (in /export/home/ssikorsk/dv
==30773==    by 0x44A7D3: WorkspaceWork::WorkspaceWork() (in /export/home/ssikor
==30773==    by 0x45C1DB: Ide::Ide() (in /export/home/ssikorsk/dvlp/upp/svn/upp/
==30773==    by 0x482D8C: GuiMainFn_() (in /export/home/ssikorsk/dvlp/upp/svn/up
==30773==    by 0x484171: main (in /export/home/ssikorsk/dvlp/upp/svn/upp/out/GC
==30773==
```

Code:

```
ScrollBar& ScrollBar::SetStyle(const Style& s)
{
 if(style != &s) {
  style = &s;
  RefreshLayout();
  Refresh();
 }
 return *this;
}
```

That "if(style != &s)" looks very suspicious to me ...

Quote:
-TreeCtrl::TreeCtrl()
+TreeCtrl::TreeCtrl() :
+ selectcount(0)

"selectcount" is initialized by Clear call in constructor.

Valgrind:

```
==31285==    at 0x672D10: Upp::TreeCtrl::Dirty(int) (in /export/home/ssikorsk/dv
==31285==    by 0x67545A: Upp::TreeCtrl::Clear() (in /export/home/ssikorsk/dvlp/
==31285==    by 0x6756A8: Upp::TreeCtrl::TreeCtrl() (in /export/home/ssikorsk/dv
==31285==    by 0x6142BF: Upp::HelpWindow::HelpWindow() (in /export/home/ssikors
==31285==    by 0x42EE02: TopicCtrl::TopicCtrl() (in /export/home/ssikorsk/dvlp/
==31285==    by 0x45C91F: Ide::Ide() (in /export/home/ssikorsk/dvlp/upp/svn/upp/
==31285==    by 0x482D8C: GuiMainFn_() (in /export/home/ssikorsk/dvlp/upp/svn/up
==31285==    by 0x484171: main (in /export/home/ssikorsk/dvlp/upp/svn/upp/out/GC
==31285==
```

Code:

```
void   TreeCtrl::Clear()
{
 item.Clear();
 item.Add();
 item[0].linei = -1;
 item[0].parent = -1;
 item[0].canopen = true;
 freelist = -1;
 Dirty();
 cursor = anchor = -1;
 selectcount = 0;
}
```

Dirty() is called before initialization of select count.

Quote:
Stopped there.... Finding them in .diff is too time consuming, maybe you could just post me the list
of problems instead of .diff?

I can post either diff or results of valgrind processing, which you can produce yourself ...

Quote:
P.S.: Do not get me wrong, valgrind is perhaps a good idea! But looks like it produces some false
alarms too...

As you can see from examples above, valgrind is not giving false alarms.

OK, thanks for clarification, you are right (and I am wrong, once again

Mirek

Subject: Re: Uninitialized class members.
Posted by mirek on Tue, 06 May 2008 19:30:42 GMT

```
 LineEdit::LineEdit() {
+    isdrag = true;
```

Why true? IMO should be false.

Mirek

Subject: Re: Uninitialized class members.
Posted by mirek on Tue, 06 May 2008 19:36:52 GMT

```
--- uppsrc/Core/Core.upp (revision 226)
+++ uppsrc/Core/Core.upp (working copy)
@@ -141,3 +141,4 @@
 mainconfig
  "Normal" = "",
  "Remote shared" = "SHARED";
+
```

I should ignore this, rigth?

Mirek

Subject: Re: Uninitialized class members.
Posted by mirek on Tue, 06 May 2008 19:42:03 GMT

Well, thank you Novo, all patches applied.

Please, continue running these diag tools, it seems really helpful

Mirek

Subject: Re: Uninitialized class members.
Posted by Novo on Tue, 06 May 2008 19:50:00 GMT

luzr wrote on Tue, 06 May 2008 15:42Well, thank you Novo, all patches applied.

Please, continue running these diag tools, it seems really helpful

Mirek

Thanks.

I'm glad to be helpful

Subject: Re: Uninitialized class members.
Posted by mdelfede on Tue, 06 May 2008 21:18:11 GMT
View Forum Message <> Reply to Message

It would be quite interesting to integrate Valgrind inside the ide....

Max

Subject: Re: Uninitialized class members.
Posted by Novo on Wed, 07 May 2008 02:10:54 GMT
View Forum Message <> Reply to Message

mdelfede wrote on Tue, 06 May 2008 17:18It would be quite interesting to integrate Valgrind inside the ide....

Max

That would be useful. Especially because there is no adequate GUI tool for valgrind at the moment. I've been using valkyrie, which is QT-based, but it doesn't work with valgrind 3.3.0. vim is pretty good in handling of valgrind's output (as usual).

I'd like to mention that valgrind is just a framework. I was running a tool called memcheck, which is based on valgrind. There are other valgrind-based tools. Most popular beside memcheck are callgrind and helgrind.

memcheck can produce XML output. So, the only thing left is to parse that XML and put data into a grid or a tree.

There is a cool example of GUI for callgrind called kcachegrind.

Subject: Re: Uninitialized class members.
Posted by mirek on Wed, 07 May 2008 07:57:15 GMT

mdelfede wrote on Tue, 06 May 2008 17:18It would be quite interesting to integrate Valgrind inside the ide....

Max

Yep, I already started checking this

Anyway, right now I see (as usual the problem with output in console apps...

Mirek

Subject: Re: Uninitialized class members.
Posted by mdelfede on Wed, 07 May 2008 12:20:02 GMT

luzr wrote on Wed, 07 May 2008 09:57mdelfede wrote on Tue, 06 May 2008 17:18It would be quite interesting to integrate Valgrind inside the ide....

Max

Yep, I already started checking this

Anyway, right now I see (as usual the problem with output in console apps...

Mirek

Which problem ?

Max

Subject: Re: Uninitialized class members.
Posted by mirek on Wed, 07 May 2008 14:32:11 GMT

mdelfede wrote on Wed, 07 May 2008 08:20luzr wrote on Wed, 07 May 2008 09:57mdelfede wrote on Tue, 06 May 2008 17:18It would be quite interesting to integrate Valgrind inside the ide....

Max

Yep, I already started checking this

Anyway, right now I see (as usual the problem with output in console apps...

Mirek

Which problem ?

Max

That the output is mixed (normal output from the app and valgrind output). Same as with gdb.

But now thinking about it, maybe this is what error output is for, right?

Mirek

Subject: Re: Uninitialized class members.
Posted by Novo on Wed, 07 May 2008 15:38:17 GMT
View Forum Message <> Reply to Message

luzr wrote on Wed, 07 May 2008 10:32
That the output is mixed (normal output from the app and valgrind output). Same as with gdb.

Options below could be useful ...


```
  --log-fd=<number>       log messages to file descriptor [2=stderr]
  --log-file=<file>       log messages to <file>
  --xml=yes               all output is in XML (some tools only)
```

Subject: Re: Uninitialized class members.
Posted by Novo on Wed, 07 May 2008 15:40:35 GMT
View Forum Message <> Reply to Message

luzr wrote on Wed, 07 May 2008 03:57mdelfede wrote on Tue, 06 May 2008 17:18It would be quite interesting to integrate Valgrind inside the ide....

Max

Yep, I already started checking this

Could you please also check on integration of callgrind?

Subject: Re: Uninitialized class members.
Posted by cocob on Wed, 07 May 2008 15:55:56 GMT
View Forum Message <> Reply to Message

Oh yes !, i'am really interested by these features !

Subject: Re: Uninitialized class members.
Posted by Novo on Wed, 07 May 2008 17:51:11 GMT
View Forum Message <> Reply to Message

luzr wrote on Tue, 06 May 2008 15:42Well, thank you Novo, all patches applied.

Please, continue running these diag tools, it seems really helpful

Mirek

Another diff-file.

Sometimes it is really hard to find out which member is not initialized, so I moved initialization of all class members to initialization list instead of assignment in many cases.

Opening TheIDE and loading a project looks almost clean now.

```
File Attachments
1) uninitialized_members_2.diff, downloaded 362 times
```

Subject: Re: Uninitialized class members.
Posted by mirek on Wed, 07 May 2008 20:47:16 GMT
View Forum Message <> Reply to Message

-ScrollBar::ScrollBar() {

```
+ScrollBar::ScrollBar()
+: style(NULL)
+, pagepos(0)
+, pagesize(0)
+, totalsize(0)
+{
  minthumb = 16;
- pagepos = pagesize = totalsize = 0;
  linesize = 1;
```

?

Same for ArrayCtrl, RichTextView, TopWindow, Ide, Browser....

Mirek

Subject: Re: Uninitialized class members.
Posted by Novo on Wed, 07 May 2008 21:05:17 GMT

I personally would move all initialization of members into initialization lists, if I understand you correctly.

I traced only several classes. And it took me two or three hours to understand what is wrong with ArrayCtrl. It is just easier to use initialization lists from very beginning.

BTW, helgrind doesn't complain about TheIDE.

Subject: Re: Uninitialized class members.
Posted by mdelfede on Wed, 07 May 2008 22:06:43 GMT

luzr wrote on Wed, 07 May 2008 16:32mdelfede wrote on Wed, 07 May 2008 08:20luzr wrote on Wed, 07 May 2008 09:57mdelfede wrote on Tue, 06 May 2008 17:18It would be quite interesting to integrate Valgrind inside the ide....

Max

Yep, I already started checking this

Anyway, right now I see (as usual the problem with output in console apps...

Mirek

Which problem ?

Max

That the output is mixed (normal output from the app and valgrind output). Same as with gdb.

But now thinking about it, maybe this is what error output is for, right?

Mirek

Uhmmmm.... I don't know, but I guess there should be a way to redirect GDB/Valgrind output to a different stream than app output... or, you could attach user app output to a different stream than stdout/stderr.
I'll investigate that one, maybe

Max

Subject: Re: Uninitialized class members.
Posted by Novo on Wed, 07 May 2008 22:21:52 GMT

mdelfede wrote on Wed, 07 May 2008 18:06
Uhmmmm.... I don't know, but I guess there should be a way to redirect GDB/Valgrind output to a different stream than app output... or, you could attach user app output to a different stream than stdout/stderr.
I'll investigate that one, maybe

Max

As I wrote previously:

    --log-fd=<number>   log messages to file descriptor [2=stderr]
    --log-file=<file>   log messages to <file>
    --xml=yes   all output is in XML (some tools only)

Subject: Re: Uninitialized class members.
Posted by mirek on Wed, 07 May 2008 23:00:14 GMT

View Forum Message <> Reply to Message

Novo wrote on Wed, 07 May 2008 17:05luzr wrote on Wed, 07 May 2008 16:47
-ScrollBar::ScrollBar() {
+ScrollBar::ScrollBar()
+: style(NULL)
+, pagepos(0)
+, pagesize(0)
+, totalsize(0)
+{
  minthumb = 16;
- pagepos = pagesize = totalsize = 0;
  linesize = 1;

?

Same for ArrayCtrl, RichTextView, TopWindow, Ide, Browser....

Mirek

I personally would move all initialization of members into initialization lists, if I understand you correctly.

OK, I will be thinking about it, although I do not find it particulary atractive... Anyway, for now, I just wanted to clarify the problem:)

Mirek

Page 13 of 17 ---- Generated from   U++ framework

Subject: Re: Uninitialized class members.
Posted by Novo on Thu, 08 May 2008 03:37:16 GMT

luzr wrote on Wed, 07 May 2008 19:00OK, I will be thinking about it, although I do not find it particulary atractive... Anyway, for now, I just wanted to clarify the problem:)

Mirek

The problem is that when you want to run valgrind on TheIDE to understand why it is crashing in release build, you see thousands of messages about using uninitialized memory.

It is just impossible to find something useful among them. Each of them can be a problem.

And of course, there are other concerns ...

Subject: Re: Uninitialized class members.
Posted by mirek on Thu, 08 May 2008 06:00:48 GMT

I think the question was whether valgrind detected a problem in ArrayCtrl (and others) when initialization was in constructor body vs in initializer list.

Anyway, I hope I will try myself today...

Mirek

Subject: Re: Uninitialized class members.
Posted by Novo on Fri, 09 May 2008 02:37:23 GMT

luzr wrote on Thu, 08 May 2008 02:00I think the question was whether valgrind detected a problem in ArrayCtrl (and others) when initialization was in constructor body vs in initializer list.

Sorry, I misunderstood you.

Yes, it can detect that. If an uninitialized variable was used in an arithmetic expression, and that expression was used in a logical expression after that, you will be warned. Actually, it is really hard to figure out which exactly variable was that. It is much easier to put all members onto initialization list.

valgrind is very important tool to me, and I'd like to see U++ valgrind-frendly, and I'm ready to help.

Besides, it helps find bugs.

Subject: Re: Uninitialized class members.
Posted by mirek on Sun, 11 May 2008 16:01:17 GMT

Novo wrote on Thu, 08 May 2008 22:37luzr wrote on Thu, 08 May 2008 02:00I think the question was whether valgrind detected a problem in ArrayCtrl (and others) when initialization was in constructor body vs in initializer list.

Sorry, I misunderstood you.

Yes, it can detect that. If an uninitialized variable was used in an arithmetic expression, and that expression was used in a logical expression after that, you will be warned. Actually, it is really hard to figure out which exactly variable was that. It is much easier to put all members onto initialization list.

valgrind is very important tool to me, and I'd like to see U++ valgrind-frendly, and I'm ready to help.

Besides, it helps find bugs.

Well, I believe theide and U++ is now quite valgrind friendly.

Anyway, I am seeing a lot of "invalid read" entries, but none seems to originate from U++ code (they seem to be "ld", "glibc" and "gtk" issues). What to think about it?

(Of course, valgrind reports a lot of leaks, but that is the same problem IMO; there should be no leaks in U++ code).

Mirek

Subject: Re: Uninitialized class members.
Posted by Novo on Sun, 11 May 2008 18:43:05 GMT

luzr wrote on Sun, 11 May 2008 12:01Well, I believe theide and U++ is now quite valgrind friendly.

Thank you very much!

Quote:

Anyway, I am seeing a lot of "invalid read" entries, but none seems to originate from U++ code (they seem to be "ld", "glibc" and "gtk" issues). What to think about it?

The is nothing you can do about these "ld", "glibc" and "gtk" issues. Valgrind allows you to provide a suppression file. Suppressions can be generated by valgrind automatically (in interactive mode).

I personally never use valgrind without a customized suppression file.

Quote:
(Of course, valgrind reports a lot of leaks, but that is the same problem IMO; there should be no leaks in U++ code).

It depend on what you call a leak. As I remember, valgrind reports a lot of "potentially lost memory". That usually means that there is non-freed memory on application exit. People usually do not care about that memory because their application is going to stop working anyway, but valgrind does.

These "memory leaks" are usually fixed by eliminating static objects (like strings) and registering memory pools cleanups with atexit().

I'll take a look at that when I get a chance.

Thank you for integrating TheIDE with valgrind again!

Subject: Re: Uninitialized class members.
Posted by mirek on Sun, 11 May 2008 20:56:52 GMT
View Forum Message <> Reply to Message

Novo wrote on Sun, 11 May 2008 14:43
It depend on what you call a leak. As I remember, valgrind reports a lot of "potentially lost memory". That usually means that there is non-freed memory on application exit. People usually do not care about that memory because their application is going to stop working anyway, but valgrind does.

Actually, U++ will complain there too, unless you use "MemoryAllocPermanent"... (or activate leak supression block).

Mirek

Subject: Re: Uninitialized class members.

Posted by [mdelfede](#) on Mon, 19 May 2008 20:23:30 GMT

BTW, Novo.... don't you have a ready-to-use valgrind suppression file, in order to avoid thousands of useless reports about internal gnome/gtk/x11 problems ?

Max

Subject: Re: Uninitialized class members.
Posted by [Novo](#) on Mon, 19 May 2008 21:20:21 GMT

mdelfede wrote on Mon, 19 May 2008 16:23BTW, Novo.... don't you have a ready-to-use valgrind suppression file, in order to avoid thousends of useless reports about internal gnome/gtk/x11 problems ?

Max

I'll prepare one for X11 + Ubunu 7.10. Unfortunately I cannot build against GTK because of a glitch in TheIDE in release mode, which always compiles static version instead of DLL-based. :-/

Subject: Re: Uninitialized class members.
Posted by [mdelfede](#) on Tue, 20 May 2008 11:44:51 GMT

Thank you !
I think Valgrind can be really useful then!

Max

Subject: Re: Uninitialized class members.
Posted by [Novo](#) on Fri, 30 May 2008 04:06:44 GMT

http://www.ultimatepp.org/forum/index.php?t=msg&goto=161 63

No GTK at the moment. Only X11.