Subject: Untrivial EditString bug(?) in FreeBSD(and may be all POSIX)
Posted by Mindtraveller on Wed, 01 Apr 2009 21:21:47 GMT
View Forum Message <> Reply to Message

I develop some app for both Windows and FreeBSD systems. Last code changes revealed very untrivial (from my point of view) behaviour in POSIX systems. When compiled and linked, application is executed in Windows flawlessly without any runtime errors. In FreeBSD the same application causes segmentation fault on startup.
I dag into the source of this fault.
So the fault happens when static EditString class member is appeared.
It looks like this:

```
//.h
class AAA {static EditString es;};
//.cpp
//...
EditString AAA::es;
```

Debugger shows call stack when this fault happens:

```
Quote:XInternAtom () from /usr/local/lib/libX11.so.6
Upp::XAtom ()
Upp::Ctrl::IsCompositedGui ()
Upp::Ctrl::Ctrl ()
Upp::EditField::EditField ()
Upp::EditValue<Upp::WString, Upp::ConvertString>::EditValue ()
Upp::EditString::EditString ()
```

If I make this EditString non-static, application runs flawlessly without faults. It seems like something in EditString ctor causes big segmentation fault when executed on program start, in the globals initialization period.

Subject: Re: Untrivial EditString bug(?) in FreeBSD(and may be all POSIX)
Posted by Mindtraveller on Sat, 04 Apr 2009 17:21:19 GMT
View Forum Message <> Reply to Message

I`ve made trivial testcase:

```
#include <CtrlLib/CtrlLib.h>
using namespace Upp;

class TestWindow : public TopWindow
{
};

EditString es; //if you comment it, everything is OK

GUI_APP_MAIN
{
 TestWindow().Run();
```

}

Subject: Re: Untrivial EditString bug(?) in FreeBSD(and may be all POSIX)
Posted by mirek on Sat, 04 Apr 2009 18:09:34 GMT

View Forum Message <> Reply to Message

Mindtraveller wrote on Sat, 04 April 2009 13:21I`ve made trivial testcase:

```
#include <CtrlLib/CtrlLib.h>
using namespace Upp;

class TestWindow : public TopWindow
{
};

EditString es; //if you comment it, everything is OK

GUI_APP_MAIN
{
 TestWindow().Run();
}
```

Because es is global.

For some untrivial subtle reasons, it is not possible to have widgets as global variables (they get initialized too soon).

Mirek

Subject: Re: Untrivial EditString bug(?) in FreeBSD(and may be all POSIX)
Posted by Mindtraveller on Sat, 04 Apr 2009 20:39:31 GMT

View Forum Message <> Reply to Message

IMO this fact [slightly] violates "everything belongs somewhere" rule, which is one of the mains of U++.
Maybe it will be wiser to have everything possible to be static (global), but the real initialization should be postponed to the first draw/paint attempt? Of maybe it would be wiser to have internal controls list, which is initialized when it is possible?

In any way, this is crucial development information. And IMO it should be resided somewhere to be seen by everyone (i.e. in the beginning of GUI tutorial).

Subject: Re: Untrivial EditString bug(?) in FreeBSD(and may be all POSIX)
Posted by mirek on Sun, 05 Apr 2009 12:50:43 GMT

View Forum Message <> Reply to Message

Mindtraveller wrote on Sat, 04 April 2009 16:39IMO this fact [slightly] violates "everything belongs somewhere" rule, which is one of the mains of U++.
Maybe it will be wiser to have everything possible to be static (global), but the real initialization should be postponed to the first draw/paint attempt? Of maybe it would be wiser to have internal controls list, which is initialized when it is possible?

In any way, this is crucial development information. And IMO it should be resided somewhere to be seen by everyone (i.e. in the beginning of GUI tutorial).

I agree with both. I too consider it rather a flaw; however it is one that is harder to fix while in practice fixing it would bring only very little benefits.

Mirek

Subject: Re: Untrivial EditString bug(?) in FreeBSD(and may be all POSIX)
Posted by mirek on Sun, 05 Apr 2009 12:53:29 GMT

View Forum Message <> Reply to Message

Added to U++ traps and pitfals...

Mirek

Subject: Re: Untrivial EditString bug(?) in FreeBSD(and may be all POSIX)
Posted by Mindtraveller on Sun, 05 Apr 2009 21:22:36 GMT

View Forum Message <> Reply to Message

Thank you!

Mirek, could you please tell what is your typical solution to avoid static controls? How do you then manage controls shared between a number of objects of the same class.

To move this question into more practical area, I`ll post the exact situation.

Application works with a scheme. Scheme consists of scheme elements. Elements are of different types. Clicking some element on the scheme drawn shows edit control in the toolbar. And the type of this controls depends on element type.
So I have a number of SchemeElement descendants
SchemeElementAAA : SchemeElement
SchemeElementBBB : SchemeElement
SchemeElementCCC : SchemeElement
and each of these classes has shared control (which is of course static).

There is of course a number of possible solutions, but IMO it would be wiser to hear one from a man who solved it many times before.

Subject: Re: Untrivial EditString bug(?) in FreeBSD(and may be all POSIX)
Posted by mirek on Thu, 21 May 2009 14:48:45 GMT

View Forum Message <> Reply to Message

Mindtraveller wrote on Sun, 05 April 2009 17:22Thank you!

Mirek, could you please tell what is your typical solution to avoid static controls? How do you then manage controls shared between a number of objects of the same class.

Lazy intitalization:

```
EditString& GlobalEditor()
{
   static EditString x;
   return x;
}
```

(Sorry for the very late reply

Mirek

Subject: Re: Untrivial EditString bug(?) in FreeBSD(and may be all POSIX)
Posted by Mindtraveller on Fri, 22 May 2009 12:11:33 GMT

View Forum Message <> Reply to Message

Simple & effective. Thanks!