Subject: Crash on Cout with MSC15
Posted by NilaT on Wed, 20 Jul 2016 14:12:27 GMT
View Forum Message <> Reply to Message

Hello,

// edit:
For those who don't want to read all the text below:
What changes have to be done to 2014 sources that MT flag will work with MSC15 like newest (2015) sources do???
Thanks

I formated my Laptop recently and decided to install the newest version of Upp and MS VS afterwards.
So I installed Upp 2015.2 (Version 9251) and MS Visual Studio Community 2015 (which is free to download).
I work on a Windows 10 Pro N Laptop.

So, after a few tweaks here and there, everything seems to work fine (despite the Win XP incompatibility, but thats okay)
BUT, I discovered one error which drives me nuts...

Cout() doesn't seem to work properly... printf is no problem though.
Sure I could change all Cout() calls to printf but that's not a solution.

The funny thing is, in debug everything works fine, only in optimal the program crashes immediately.
I tried to debug the optimal version in Visual Studio, the error seems to be in:

```
void Stream::Put(const char *s)
{
 while (*s) Put(*s++);
}
```

The error is always the same: (The error is german and I tried to translate it to english as good as I can)
Exception at 0x00AB60C7 (address changes every time) in (Projectname).exe: 0xC0000005: Access violation at read at position 0x0000004.

But the funniest thing is, that this must be something depending on my includes (it's a rather big project)
Because, I made a new test application, only containing the following code:

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

using namespace Upp;

CONSOLE_APP_MAIN
```

```
{
 Cout() << "\n---------------------------------\n";
 ReadStdIn();
}
```
and this just works fine, using the SAME sources as the big project.
The \n----... is also the first line in my other project.

Any suggestions?
Any help is very appreciated!

I also tried differend compiler flags (GS) and sources (2014, 2015), always the same result...
Everything works, except my "old" project.

//edit: I narrowed it down to the MT flag... Please just create a new multithreaded console project with the content I posted above and try it... It will crash. But why?

//edit2: Okay, only 2014 sources are affected. The newest sources work fine with MT... I'm on it.
Please keep this thread if other people have the same issue

Subject: Re: Crash on Cout with MSC15
Posted by deep on Wed, 20 Jul 2016 14:34:59 GMT
View Forum Message <> Reply to Message

Hi,

In new release "optimal" mode is deprecated.
Quote:The funny thing is, in debug everything works fine, only in optimal the program crashes immediately.
I tried to debug the optimal version in Visual Studio, the error seems to be in:

Check flags for "release" mode.

Or check with "Optimized speed" mode. I faced similar issues with optimal mode.

Subject: Re: Crash on Cout with MSC15
Posted by NilaT on Wed, 20 Jul 2016 14:43:39 GMT
View Forum Message <> Reply to Message

Thanks for the quick answer,

as I discovered, it has something to do with MT flag and the changes from 2014 sources to 2015.
Because the crash only appears in MT with 2014 sources.
MT on 2015 work fine...

I think there must be a significant difference... But where?!

Must be something about that encoding and converting stuff, right?

-

Which flags do you mean exactly, because there are sooo many different flag and option menus in Upp.
But if you mean the Setup > Build options, I got:
Common options: /bigobj /D_ATL_XP_TARGETING /FS
Debug options: -Od
Debug link options: /STACK:20000000
Optimize for speed: -O2
Optimize for size: -O1
Release link options: same as Debug link options

Subject: Re: Crash on Cout with MSC15
Posted by deep on Thu, 21 Jul 2016 05:24:20 GMT
View Forum Message <> Reply to Message

Hi,

Please check when this gets called first time

void Stream::Put(const char *s)

pointer points to valid "cstring"

Subject: Re: Crash on Cout with MSC15
Posted by NilaT on Thu, 21 Jul 2016 06:25:28 GMT
View Forum Message <> Reply to Message

Could somebody please confirm the following?
New Console App in latest upp with latest sources, MSC15 on Win 10.
Compile and run in optimal with MT flag!
Code:

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

using namespace Upp;

CONSOLE_APP_MAIN
{
 CoutUTF8(); // Without this line, works perfect. With this line = crash (same error as with 2014 upp sources)
 Cout() << "\n---------------------------------\n";
 ReadStdIn();
}
```

And explain me where and why?
Thanks.

//edit: Because we work with 2014 sources in our project, please forget that 2015 example.
I tried to apply the changes in CoutStream from 2015 to 2014 sources, because I thought a wrong Charset causes the error.
Afterwards I edited the Stream::Put method:

```
void Stream::Put(const char *s)
{
 while (*s)
 {
  Put(*s);
  *s++;
 }
}
```

In debug, Put calls void Put(int c) in Stream.h:

```
void      Put(int c)
{
  if(ptr < wrlim)
    *ptr++ = c;
  else
    _Put(c);
}
```

and then calls _Put, which, in endeffect calls the "new" Put0 which converts the charset.

BUT, in optimal, the error raises in Stream::Put -> Put call.
I debug in Visual Studio, put a Breakpoint on the Put(*s); call and press F11, but I can't jump into the function...

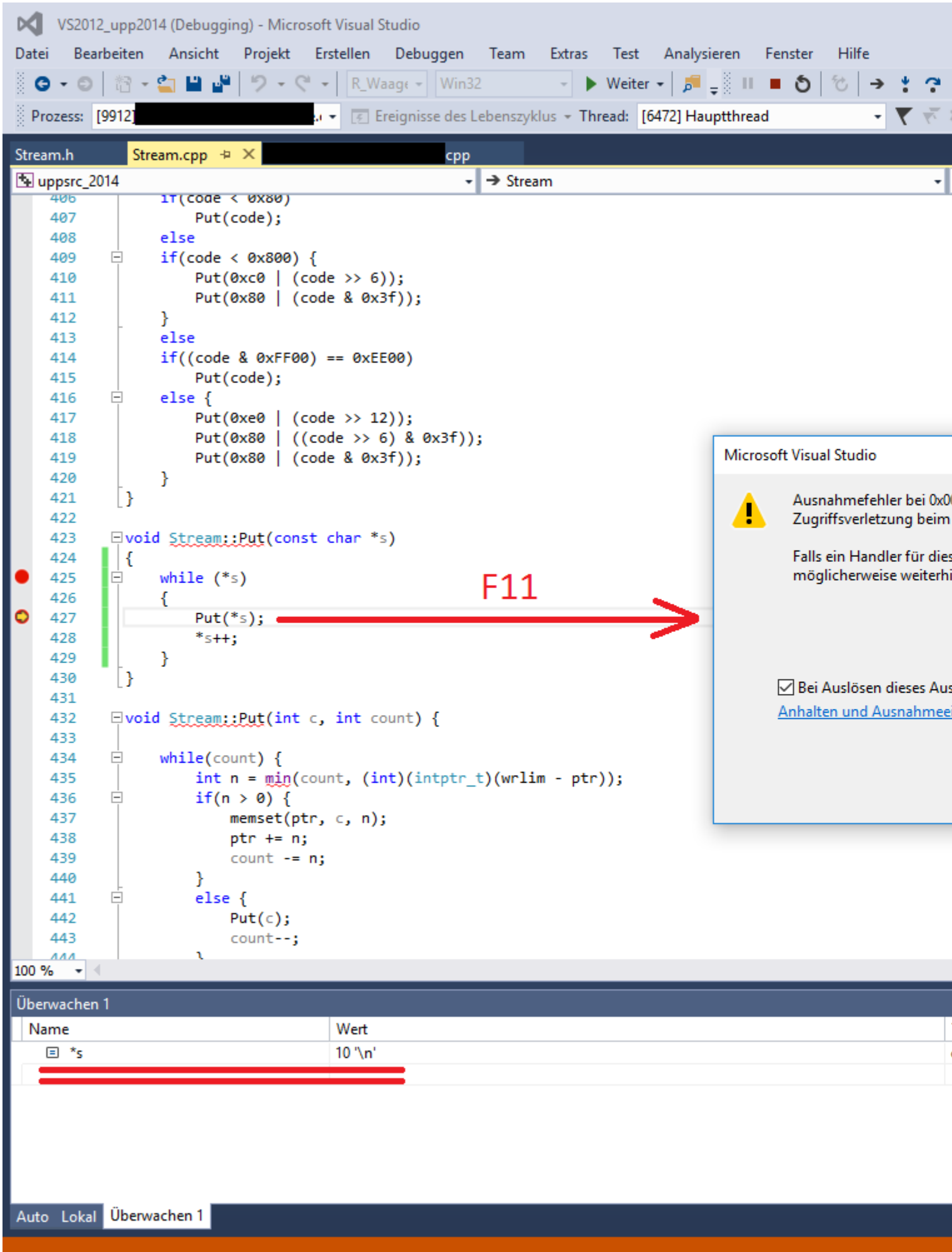
What the heck...

// edit:
I tried to compile our project with 2015 sources, same result.
Debug runs fine, optimal crashes.

I attach a pick of our project, compiled with 2014 sources and MSC15 in optimal.

File Attachments

1) crash.png, downloaded 692 times


VS2012_upp2014 (Debugging) - Microsoft Visual Studio
Datei Bearbeiten Ansicht Projekt Erstellen Debuggen Team Extras Test Analysieren Fenster Hilfe
Weiter
Prozess:
Ereignisse des Lebenszyklus
Thread: [6472] Hauptthread
Stream.h
Stream.cpp
cpp
uppsrc_2014
Stream
406 if(code < 0x80)
407 Put(code);
408 else
409 if(code < 0x800) {
410 Put(0xc0 | (code >> 6));
411 Put(0x80 | (code & 0x3f));
412 }
413 else
414 if((code & 0xFF00) == 0xEE00)
415 Put(code);
416 else {
417 Put(0xe0 | (code >> 12));
418 Put(0x80 | ((code >> 6) & 0x3f));
419 Put(0x80 | (code & 0x3f));
420 }
421 }
422
423 void Stream::Put(const char *s)
424 {
425 while (*s)
426 {
427 Put(*s);
428 *s++;
429 }
430 }
431
432 void Stream::Put(int c, int count) {
433
434 while(count) {
435 int n = min(count, (int)(intptr_t)(wrlim - ptr));
436 if(n > 0) {
437 memset(ptr, c, n);
438 ptr += n;
439 count -= n;
440 }
441 else {
442 Put(c);
443 count--;
F11
Microsoft Visual Studio
Ausnahmefehler bei 0x0
Zugriffsverletzung beim
Falls ein Handler für dies
möglicherweise weiterhi
Bei Auslösen dieses Aus
Anhalten und Ausnahmee
100 %
Überwachen 1
Name
Wert
*s
10 '\n'
Auto Lokal Überwachen 1