Subject: Freezing bare application on Windows XP
Posted by qapko on Sat, 26 Dec 2009 09:05:29 GMT
View Forum Message <> Reply to Message

Hello,
I've discovered one issue in method Ctrl::GuiSleep0 (used by TopWindow::Run()). Namely, Upp application may freeze on line ExitLoopEvent().Wait() just after starting it on Windows XP. I've seen it by providing debug logging to the sources of Upp. Then I experimented and removed the line and the bug dissapeared. I don't see deeply in the internals of event loop of Upp, so I am not sure what the line is doing here at all. Is there anybody, who understands the issue and may provide some fix? The problem is appearing with my regular Upp applications also. It's sad, that Upp (or something below it?) has such an instability in the core of itself.
I'm attaching my testsuite that I've created to make the problem more frequent to see.
I also noticed that Ctrl::ProcessEvent and Ctrl::ProcessEvents have no GuiLock at the start of the methods in Windows version, but X11 version have ones. Is it OK? Can't it freeze the application while running? One of my customers is complaining that GUI of the Upp application, that we are developing for him, is freezeng occasionaly, but he can see that all the threads of the application behind GUI work fine in the situation (but the threads are written in Python, not in Upp).
I'm prepared to test the issue further with some instructions from you.
Thank you for your help.

Have a happy new year with this outstanding framework!

Gabi

File Attachments
1) kmedis.zip, downloaded 500 times

Subject: Re: Freezing bare application on Windows XP
Posted by mirek on Sat, 26 Dec 2009 13:46:25 GMT
View Forum Message <> Reply to Message

qapko wrote on Sat, 26 December 2009 04:05Hello,
I've discovered one issue in method Ctrl::GuiSleep0 (used by TopWindow::Run()). Namely, Upp application may freeze on line ExitLoopEvent().Wait() just after starting it on Windows XP. I've seen it by providing debug logging to the sources of Upp. Then I experimented and removed the line and the bug dissapeared. I don't see deeply in the internals of event loop of Upp, so I am not sure what the line is doing here at all. Is there anybody, who understands the issue and may provide some fix? The problem is appearing with my regular Upp applications also. It's sad, that Upp (or something below it?) has such an instability in the core of itself.

Unfortunately, I am not able to reproduce the bug nor I ever experienced it.... And really never encountered the issue in practive and I would say we really mainaint a lot of U++ apps... The only possible explanation is that it is somewhat connected with that initial timer call, which is a bit unusuall - but I still see no reason why it should mess the code.

Explanation of situation: There is some weird "OverWatchThread" in U++, which is basically only meant to patch exiting the application, as we had some weird issues there - namely, Windows displayed "Close" tooltip 2-3 seconds after you have closed the application with top-right corner X icon.

Digging into the issue, I have found that Win32 simply expects the appliaction to recieve WM_QUIT message as the only way to exit. Unfortunately, that is not really compatible with U++ way, where we just quit by exiting GUI_APP_MAIN. That is why we have the OverWatchThread - it is there only to "wait" for WM_QUIT message so that Win32 things everything is OK. Note that the main thread send WM_USER to this thread only to make it invoke "PostQuitMessage(0);".

This thread uses ExitEventLoop Event to communicate with the main GUI thread. It sets the Event twice - first time to announce that OverWatchTread is started and running, second time to announce that the application is exiting.

For even other reasons, namely initialisaton of .ocx (you cannot start a new thread in DllMain), Tom has moved initialisation of this thread to GuiSleep.

Just staring at the code, I soo no real reason why it should not work. Simply, there are no other complex things in OverWatchThread startup - it should just create an invisible window and set the event.

In any case, please activate ELOGs - line 14 of the file and send the output .log.

Quote:
I also noticed that Ctrl::ProcessEvent and Ctrl::ProcessEvents have no GuiLock at the start of the methods in Windows version, but X11 version have ones. Is it OK? Can't it freeze the application while running? One of my customers is complaining that GUI of the Upp application, that we are developing for him, is freezeng occasionaly, but he can see that all the threads of the application behind GUI work fine in the situation (but the threads are written in Python, not in Upp).

Should be ok. Maybe there are uncessarry GuiLocks in some cases, but they are suspended by LeaveGMutexAll(); anyway.

Mirek

Subject: Re: Freezing bare application on Windows XP
Posted by qapko on Sat, 26 Dec 2009 18:08:45 GMT
View Forum Message <> Reply to Message

Hello,
thank you for the response I'm sending you logs I've collected while testing the issue after activating ELOGs. Calling GetSysTime in ELOGs causes crashing application occasionaly - look at the date in a2.log for example. Removing GetSysTime from ELOGs causes to freeze the application only "in the old manner"

Removing SetTimeCallback from Kmedis contructor and closing the application manually makes the same results. All log files are commented at the beginning of the file.

Gabi

File Attachments

1) logs.zip, downloaded 467 times

Subject: Re: Freezing bare application on Windows XP
Posted by mirek on Sat, 26 Dec 2009 19:51:07 GMT
View Forum Message <> Reply to Message

qapko wrote on Sat, 26 December 2009 13:08Hello,
thank you for the response I'm sending you logs I've collected while testing the issue after activating ELOGs. Calling GetSysTime in ELOGs causes crashing application occasionaly - look at the date in a2.log for example. Removing GetSysTime from ELOGs causes to freeze the application only "in the old manner"
Removing SetTimeCallback from Kmedis contructor and closing the application manually makes the same results. All log files are commented at the beginning of the file.

Gabi

Some of them look like everything went ok?! E.g. A1.

Can you be more descriptive please?

Mirek

Subject: Re: Freezing bare application on Windows XP
Posted by qapko on Sat, 26 Dec 2009 20:15:07 GMT
View Forum Message <> Reply to Message

Yes,
A1 and B1 are from running the application without crashing or freezing. All As are from runs with GetSysTime in ELOGs. All Bs are from runs without GetSysTime in ELOGs. Some of them are with and some without SetTimeCallback in Kmedis contructor, which proves, that calling SetTimeCallback in Kmedis constructor has no effect on the bug.
In Bs the application hangs apparently on ExitLoopEvent().Wait() line in GuiSleep0(). I can prove it by putting RLOG before and after the line.

Gabi

Subject: Re: Freezing bare application on Windows XP
Posted by mirek on Sat, 26 Dec 2009 20:40:41 GMT
View Forum Message <> Reply to Message

Well, first partial finding:

The cause of crashing for GetSysTime is really stupid - in single threaded mode, you cannot call any U++ routine in non-main thread and ELOG does exactly that.... (in OverWatchThread).

Note: Otherwise, OverWatchThread is carefully designed not to call anything that would have this problem - it only calls win32 API which is MT safe.

BTW, have you tried this app with MT flag (it would fix GetSysTime problem).

(I have reproduced GetSysTime problem, but nothing else so far).

If you are getting consistent crashes, maybe one thing to try is to replace Event with Semaphore (or StaticSemaphore)... (Just replace Set with Release....).

BTW, you have not posted details of your compiler / system yet...

(I am trying to reproduce the problem in WinXP with MSC9).

Mirek

Subject: Re: Freezing bare application on Windows XP
Posted by mirek on Sat, 26 Dec 2009 20:53:07 GMT
View Forum Message <> Reply to Message

Thinking about it...

First of all, sorry about StaticSemaphore suggestion - that cannot work in single-threaded mode.

Which brings us to the possible reason anyway - maybe the problem is

GLOBAL_VAR(Event, Ctrl::ExitLoopEvent)

because just like anything else, in single-threaded, the GLOBAL_VAR does not provide thread protection. In other words, it can lead to the problem when two threads initialize Event twice and then work with different Event handle.

Fix is simple, use

Event Ctrl::ExitLoopEvent;

instead (change from function call to class static variable).

Can you try that please?

Mirek

Subject: Re: Freezing bare application on Windows XP
Posted by qapko on Sat, 26 Dec 2009 21:09:47 GMT
View Forum Message <> Reply to Message

Yes,
as far as now defining Ctrl::ExitLoopEvent as static attribute of the class seems to solve the problem. I will test it for some 30 minutes to be sure it's OK now. Should I do any other changes to the source then let me know please.

Gabi

Subject: Re: Freezing bare application on Windows XP
Posted by qapko on Sat, 26 Dec 2009 21:58:32 GMT
View Forum Message <> Reply to Message

After the testing I hope it's OK now. And again, thank you for your help. I highly appreciate your attention.
Just two more questions:
1. Can this concrete bug be cause of those strange freezings of our application at the customer site while the application is running (just not after the start)?
2. Can you please take a look and find, whether the issue with GLOBAL_VAR may not cause similar problems (deadlocks, namely) on another places in Upp framework? I see clearly, that with my knowledge of Upp I'm not able to do that job. But I can help you with some testing of course.

Gabi

PS: My wife is saying we are crazy, digging in the sources at Christmas time But she has some sense of the situation also. When I was teeth-brusing while running the last tests, she cryied to me suddenly: "Not responding!" Uff, she know how to bite me

---

Subject: Re: Freezing bare application on Windows XP
Posted by mirek on Sat, 26 Dec 2009 23:27:45 GMT
View Forum Message <> Reply to Message

qapko wrote on Sat, 26 December 2009 16:58After the testing I hope it's OK now. And again, thank you for your help. I highly appreciate your attention.

Just two more questions:
1. Can this concrete bug be cause of those strange freezings of our application at the customer site while the application is running (just not after the start)?

Unlikely. This is a very specific race condition that happens only during first call of GuiSleep. You must be very lucky that your machine and OS and everything else actually match requirements to trigger this issue...

But it nicely demonstrates what is so hard about MT

Quote:
2. Can you please take a look and find, whether the issue with GLOBAL_VAR may not cause similar problems (deadlocks, namely) on another places in Upp framework? I see clearly, that with my knowledge of Upp I'm not able to do that job. But I can help you with some testing of course.

Well, this is one of two places where another thread is created in single-threaded mode. Other place is much more simple.

The real problem is that GLOBAL_VAR definition differes in both modes - in ST it is made simpler, bypassing all MT stuff. Which is bug here, but OK in normal use.

Thanks for making me aware about this problem.

Mirek

Subject: Re: Freezing bare application on Windows XP
Posted by mirek on Sat, 26 Dec 2009 23:30:29 GMT
View Forum Message <> Reply to Message

I have commited the fix, please check the final fix ASAP.

Mirek

Subject: Re: Freezing bare application on Windows XP
Posted by qapko on Sun, 27 Dec 2009 07:48:07 GMT
View Forum Message <> Reply to Message

I've tested the SVN version. It's OK.

Gabi

Subject: Re: Freezing bare application on Windows XP
Posted by mirek on Sun, 27 Dec 2009 18:49:23 GMT
View Forum Message <> Reply to Message

qapko wrote on Sun, 27 December 2009 02:48I've tested the SVN version. It's OK.

Gabi

Uff, thanks. Nasty stuff....

Mirek

Subject: Re: Freezing bare application on Windows XP
Posted by qapko on Sun, 27 Dec 2009 20:31:58 GMT
View Forum Message <> Reply to Message

In fact, I am now thinking about the best way how to eventually catch such kind of problems in Upp. We are developing embedded systems for data acquisition and the Upp application is just one piece of the whole system. But at some sites it's the only way how to see some real data from the field. It's not critical to keep the application running all the time, because all the data are stored at lower levels below the operator's PC, but from customer's view it's bad when the software freezes would it be just once for half an year.
Don't you have some set of stress tests which I may run on my PC to attract those kind of problems?
I would like to make sure that Upp is the rock solid part of my aplications, the other would be the Python (hopefuly) - leaving all the blame of any instability of my application on me, and also simplifying the bug hunting of course

Have a nice day

Gabi

Subject: Re: Freezing bare application on Windows XP
Posted by mirek on Tue, 29 Dec 2009 13:14:05 GMT
View Forum Message <> Reply to Message

qapko wrote on Sun, 27 December 2009 15:31In fact, I am now thinking about the best way how to eventually catch such kind of problems in Upp. We are developing embedded systems for data acquisition and the Upp application is just one piece of the whole system. But at some sites it's the only way how to see some real data from the field. It's not critical to keep the application running all the time, because all the data are stored at lower levels below the operator's PC, but from customer's view it's bad when the software freezes would it be just once for half an year.
Don't you have some set of stress tests which I may run on my PC to attract those kind of problems?
I would like to make sure that Upp is the rock solid part of my aplications, the other would be the Python (hopefuly) - leaving all the blame of any instability of my application on me, and also

simplifying the bug hunting of course

Have a nice day

Gabi

It is a bit hard to predict problems like the one we have just fixed.

Where possible, we have used extensive automated testing in parts that are very complex and error-prone (memory allocator, String etc...). You can see some of the code in trunk\upptst

---

Subject: Re: Freezing bare application on Windows XP
Posted by qapko on Wed, 30 Dec 2009 07:50:17 GMT
View Forum Message <> Reply to Message

Thank you for all the informations you have provided to me. I've never looked at the sources of Upp except while hunting some bug or studying some less documented functionality. But as for bugs, Upp have given me too little opportunities to look at the code at all I think it's worth to study it so that I have some sense of how things are running in th core. I can test my applications more thouroughly then, I hope. Once more, thank you for your support.
Have a nice day

Gabi
