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