Subject: Re: Kill callbacks when parent is destructed
Posted by crydev on Thu, 04 Sep 2014 07:25:41 GMT
View Forum Message <> Reply to Message

mirek wrote on Tue, 02 September 2014 18:44crydev wrote on Tue, 02 September 2014 18:04Hello,

I have a question about callbacks. I'm using a TopWindow that creates asynchronous callbacks. When the TopWindow is closed, it may be possible that one of the callbacks are still being executed. I changed the declaration to PTEBACK2 instead of THISBACK2, which seemed to help for most cases. However, I still experience problems. Now I know that you cannot just terminate an execution path. I could build in the necessary synchronization. However, when I saw the PTEBACK alternative, I was thinking: Does U++ provide a method to prevent crashing when execution PTEBACK's when its parent TopWindow is closed a.k.a. destroyed? Is there a simpler solution I can build in to suppress the situation?

Thanks,

crydev

Not sure I understand all detail properly, but when it is possible that parent TopWindow is destroyed _while_ _executing_ Callback, then you have violated multithreading rules (whole GUI is shared resource, only one thread is allowed to run GUI and some operations like opening/closing windows are allowed only for the main thread).

Thanks for your reply Mirek,

The callbacks I am executing run on seperate threads, using Thread().Run(...). This way, I can close the window without affecting the running threads. I am not operating on the user interface from other threads. I assume that scheduled PostCallback's are automatically removed when the TopWindow is closed, so one of my threads should be causing the problem.

Thanks

crydev