
Subject: MT opens up whole new can of worms :)
Posted by [cbpporter](#) on Wed, 16 Mar 2016 13:03:09 GMT
[View Forum Message](#) <> [Reply to Message](#)

So I started doing heavy MT for all my IDE related tasks, and man, nothing works as it did anymore. All the GUI conventions are thrown on their back, like when you edit some code, which launches a thread, which eventually finishes while you edit, which differentially updates the code navigator, re-parses documentation comments, all while triggering events left and right, having unexpected cursor change and content edits events.

Do you guys do MT GUI?

Subject: Re: MT opens up whole new can of worms :)
Posted by [mirek](#) on Wed, 16 Mar 2016 16:16:49 GMT
[View Forum Message](#) <> [Reply to Message](#)

cbpporter wrote on Wed, 16 March 2016 14:03So I started doing heavy MT for all my IDE related tasks, and man, nothing works as it did anymore. All the GUI conventions are thrown on their back, like when you edit some code, which launches a thread, which eventually finishes while you edit, which differentially updates the code navigator, re-parses documentation comments, all while triggering events left and right, having unexpected cursor change and content edits events.

Do you guys do MT GUI?

I guess there is a good reason why majority of frameworks has "GUI thread" concept

Mirek

Subject: Re: MT opens up whole new can of worms :)
Posted by [koldo](#) on Thu, 17 Mar 2016 08:39:56 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello cbpporter

In my case I am rather careful: Threads do not change GUI directly, and only callback functions set in thread called with `SetTimeCallback()` monitor every certain time the changes and show them in the GUI.

An additional advantage is that this way the "algorithm" side and the "GUI" side are more isolated, so in some cases the same "algorithm" code/class is called in a command line program and in a GUI one, with no changes.

Subject: Re: MT opens up whole new can of worms :)

Posted by [cbpporter](#) on Mon, 21 Mar 2016 09:41:32 GMT

[View Forum Message](#) <> [Reply to Message](#)

Well, I do all the GUI stuff on the main thread.

Still, after years of experience with GUI, you get a very good feel of when and why events should trigger and you learn to write short code that best uses these rules.

With MT, events trigger all around in non deterministic orders and the only good solution I found is to turn some off while background threads are running and turn them back on once they are done, combined with a ticketing system, so that the event can't be triggered only if the tickets match.

One rather simple example is editing code vs. updating a code navigator. If you do it with ST, once the code analysis is done, the navigator can be updated, which will sometimes resolve in an order change of items and the cursor position must be changed. So an event will be triggered and the cursor in the editor changed, but this is not a problem since the cursor is restored after the navigation update procedure. With MT, if the cursor changes due to a thread, events shouldn't trigger since that makes the editor incapable of doing mouse select, but if you click on it, it should work.

The rest of the cases are more complicated than this.
