
Subject: Major redesign of MT in GUI

Posted by [mirek](#) on Sat, 16 May 2009 17:40:59 GMT

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

Up to now, the only allowed interaction with non-main threads and GUI was through sending events.

The new approach introduces "Ctrl::EnterMutex" and "Ctrl::LeaveMutex" methods that can be used to protect shared access to any widget (and helper class Ctrl::Lock that does so on for block using constructors/destructors).

Means accessing widget data or changing widget content in non-main thread is now much more simple.

For now (maybe only today , there are certain methods that cannot be called by non-main thread - those that deal with opening new windows and events (event loops). It is because in Win32 it is impossible (for U++) to run them in any other thread than main.

I will probably resolve this situation by performing such requests in main thread.... (via new planned "Ctrl::Call function that will make possible to 'call' code in main thread while waiting for its completion).

Mirek

Subject: Re: Major redesign of MT in GUI

Posted by [kodos](#) on Sun, 17 May 2009 10:43:59 GMT

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

If there is no performance hit, this sounds really interesting!

Subject: Re: Major redesign of MT in GUI

Posted by [mirek](#) on Sun, 17 May 2009 12:40:46 GMT

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

kodos wrote on Sun, 17 May 2009 06:43 If there is no performance hit, this sounds really interesting!

I believe there is almost no performance hit as long as most of GUI is performed in main thread

But of course, all of that needs some locking.

Mirek

Subject: Re: Major redesign of MT in GUI
Posted by [Mindtraveller](#) on Sun, 17 May 2009 14:30:54 GMT
[View Forum Message](#) <> [Reply to Message](#)

Does it mean tha each Ctrl will have it`s own Mutex object?

Subject: Re: Major redesign of MT in GUI
Posted by [mirek](#) on Sun, 17 May 2009 21:11:40 GMT
[View Forum Message](#) <> [Reply to Message](#)

Actually, there is single mutex for everything. That is required by X11 anyway and semi-required by Win32.

In fact, after some thinking, I have decided to scratch the idea of "per widget" locking and replace this global mutex available.

Also, Ctrl::Call is now implemented and thus you can absolutely anything in threads with GUI (but some things are being performed in main thread using Call).

So the final rule for MT GUI programming is pretty simple:

If you are going to call any method or GUI function, you have to lock the scope using GuiLock helper guard (or EnterGuiMutex/LeaveGuiMutex pair).

Mirek
