
Subject: Re: How do I create a loop for a window to react to volatile changes to a global variable?

Posted by [Oblivion](#) on Mon, 11 Mar 2019 11:44:22 GMT

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

Hello slashupp,

If you are calling a method of the main window (main thread), then add a GuiLock "in" that method, it is better that way.

E.g.:

```
#include <CtrlLib/CtrlLib.h>
```

```
using namespace Upp;
```

```
class ThreadTest : public TopWindow {
    Button  start;
    ArrayCtrl array;
public:
    ThreadTest()
    {
        SetRect(0, 0, 800, 600);
        CenterScreen();
        array.AddColumn("Number");
        Add(array.HSizePos().VSizePos(0, 30));
        Add(start.SetLabel("Start Thread").RightPos(2, 120).BottomPos(2, 26));

        start << [=] { RunThread(); };
    }
    void RunThread()
    {
        Thread t;
        t.Run([=]{
            for(int i = 1; i < 30000; i++) {
                // GuiLock __;
                SetArray(i);
            }
        });
        t.Detach();
    }
    void SetArray(int n)
    {
        GuiLock __;
        array.Add(n);
        if(n == 10) { // Resize window from within thread...
            SetRect(0, 0, 640, 480);
        }
    }
}
```

```
}  
  
};  
  
GUI_APP_MAIN  
{  
  ThreadTest().Run();  
}
```

Also,

Quote:

I added "GuiLock ___;" in the methods but then get:

"Assertion failed in ../CtrlCore/GtkCapture.cpp, line 41 IsMainThread()"

AFAIK, this means that, that specific method -or other methods called within it- can only be called from the main thread.

In some cases you may want to create a customized event loop in the main thread.

E.g:

```
void Run()  
{  
  OpenMain();  
  while(IsOpen()) {  
    ProcessEvents();  
    GuiSleep(10);  
    ProcessMyThreadsEvents(); // It's up to you how to process your threads' events.  
  }  
}
```

Best regards,
Oblivion
