
Subject: ProgressIndicator issue

Posted by [idkfa46](#) on Tue, 24 Mar 2020 10:21:25 GMT

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

Hallo guys,

I have a problem with my ProgressIndicator included in StatusBar.

My ProgressIndicator do not show the indicator increase till the end of the cicle (where displaying 100%)... it seems like the statusbar is not refreshed everytime, isn't it?

If I add a progress too, it works perfectly! Why?

How can I fix it removing the progress?

Here is a little example:

```
class test : public WithtestLayout<TopWindow> {
    StatusBar status;
    ProgressIndicator pi;

public:
    void load();
    typedef test CLASSNAME;
    test();
};

test::test()
{
    CtrlLayout(*this, "Window title");

    AddFrame(status);
    pi.Hide();
    pi.Percent();
    status.Add(pi.RightPos(5, 200).TopPos(2, 15));

    goBtn <<= THISBACK(load);
}

void test::load()
{
    Progress p;
    p.Create();

    pi.Show();
    pi.Set(0,1000000);
}
```

```
for(int i = 0; i <=1000000; i++)
{
    pi.Set(i);
    p.SetPos(i);
}
p.Close();
pi.Close();
}
```

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

Subject: Re: ProgressIndicator issue
Posted by [Xemuth](#) on Tue, 24 Mar 2020 11:39:40 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello ldkfa46,

Put this at begining of Load function :

```
GuiLock ___;
```

Like that's :

```
void test::load()
{
    GuiLock ___;
    Progress p;
    p.Create();

    pi.Show();
    pi.Set(0,1000000);

    for(int i = 0; i <=1000000; i++)
    {
        pi.Set(i);
        p.SetPos(i);
    }
    p.Close();
    pi.Close();
}
```

It Should allow the thread created by callback to interact with the GUI wich is stick to the main thread

Subject: Re: ProgressIndicator issue
Posted by [idkfa46](#) on Tue, 24 Mar 2020 12:02:56 GMT
[View Forum Message](#) <> [Reply to Message](#)

I tried but nothing changed

thanks for your support,
Matteo

Subject: Re: ProgressIndicator issue
Posted by [Xemuth](#) on Tue, 24 Mar 2020 12:44:04 GMT
[View Forum Message](#) <> [Reply to Message](#)

I will share you a simple test case to see

Subject: Re: ProgressIndicator issue
Posted by [Xemuth](#) on Tue, 24 Mar 2020 13:21:39 GMT
[View Forum Message](#) <> [Reply to Message](#)

Here we go, this example show you how I deal with Progress Indicator, may there is another way of doing it (without thread maybe ?) But I don't know it.

May more experienced member wich know how to deal with it in a different way can share with us.

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

```
using namespace Upp;
```

```
class testIndicator : public TopWindow {
```

```
private:
```

```
    ProgressIndicator progress;
```

```
    Button button;
```

```
public:
```

```
    typedef testIndicator CLASSNAME;
```

```
    void myCallBack(){
```

```
        Thread().Run([&](){
```

```
            GuiLock __; //Since we are in thread we cant modify GUI, Except if we create a struct Name  
GuiLock wich will give ctrl of GUI to this thread
```

```
            button.Disable(); //Action on GUI
```

```
            GuiUnlock __; // Release the GUI Ctrl (if not done then the gui can freze because main thread
```

wich processe Mouse/key Ctrl and other GUI things is stuck since it don't have Ctrl of GUI anymore)

```
for(int e = 0; e < 11; e++){
    if(Thread::IsShutdownThreads()){break;}
    GuiLock __3;
    progress = e;
    GuiUnlock __3; //It's important to release before Sleep Else the main thread wont be able to
refresh GUI during the sleep (because this thread keep the control of GUI)
    Sleep(1000);
}

GuiLock __2; //Same as before
button.Enable();
GuiUnlock __2;

return;
});
}

testIndicator(){
    Title("Exemple ProgressBar");
    SetRect(0, 0, 220, 70);

    Add(button.LeftPos(30, 150).TopPos(10, 20));
    button.SetLabel("Increase progressBar");
    button <<= THISBACK(myCallBack);

    Add(progress.LeftPos(10, 200).TopPos(40, 20));
    progress.Set(0,10);
}

~testIndicator(){
    Thread::ShutdownThreads(); //Thread released
}
};

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

Subject: Re: ProgressIndicator issue
Posted by [Xemuth](#) on Tue, 24 Mar 2020 13:28:52 GMT
[View Forum Message](#) <> [Reply to Message](#)

Also, In your exemple, the window freze because during the loop, the main thread wich is the

same as GUI thread is busy and wont refresh the GUI, that's why I have place my loop in another thread to let the GUI Thread free

Subject: Re: ProgressIndicator issue
Posted by [idkfa46](#) on Tue, 24 Mar 2020 14:41:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Thank you for your support! Your solution is really interesting but the easiest solution is to call

```
ProcessEvents();
```

as suggested me by Omari.

Best,
Matteo

Subject: Re: ProgressIndicator issue
Posted by [Xemuth](#) on Tue, 24 Mar 2020 14:56:13 GMT
[View Forum Message](#) <> [Reply to Message](#)

idkfa46 wrote on Tue, 24 March 2020 15:41 Thank you for your support! Your solution is really interesting but the easiest solution is to call

```
ProcessEvents();
```

as suggested me by Omari.

Best,
Matteo

Indeed, it work as well without a thread ! the only default of this methode is (if I put it in my exemple) the gui is still lock during the Sleep function which is normal.

Their is however many workarround like doing this kind of things :

```
void myCallBack(){  
    button.Disable(); //Action on GUI  
    ProcessEvents();  
    for(int e = 0; e < 11; e++){  
        progress = e;  
        ProcessEvents();  
        for(int i = 0; i < 100; i++){  
            Sleep(10);  
            ProcessEvents();  
        }  
    }  
    //Same as Sleep(1000) but with a better control on GUI (however it wont 'sleep' for  
    // 1 second now, but for 1 sec + 100 * (processEvents() Time)  
}
```

```
button.Enable();  
ProcessEvents();  
return;  
}
```
