
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();  
}
```
