
Subject: Re: Code before Thread.Run() nor executed
Posted by [Oblivion](#) on Tue, 15 Aug 2017 16:45:56 GMT
[View Forum Message](#) <> [Reply to Message](#)

As I suspected, you are calling GUI elements/methods from a worker thread directly. You need to use PostCallback, and make sure the thread terminates.
Here you go:

```
TestLayout::TestLayout()
{
    CtrlLayout(*this);
    button1 << THISBACK(OnClickButton);
}
```

```
void TestLayout::OnClickButton()
{
    WhenThreadStart();
}
```

```
TestLayout2::TestLayout2()
{
    CtrlLayout(*this);
    progress.Set(0, 1);
}
```

```
TestWindow::TestWindow()
{
    SetRect(0, 0, Zx(400), Zy(200));
    Add(testLayout);
    Add(testLayout2);
    testLayout.Show(true);
    testLayout2.Show(false);
    testLayout.button1.WhenAction = THISBACK(StartProgress); // button1's WhenAction callback
    was not defined.
}
```

```
void TestWindow::StartProgress()
{
    testLayout.Show(false);
    testLayout2.Show(true);
    testLayout2.progress.Set(0, 10);
    Thread().Run(THISBACK(Progress));
}
```

```
void TestWindow::Progress()
{
```

```

// This thread code could be written in a much better way, but I'm simply giving you the idea :).
for(int i = 0; i < 10; i++)
{
    if(Thread::IsShutdownThreads())
        break;
    Sleep(1000);
    PostCallback([=]{
        // Never directly call GUI elements. If you want to do fancy GUI stuff with
        // threads, use PostCallback.
        testLayout2.progress++;

    });
}
PostCallback([=]{
    // Let us roll-back;
    testLayout2.Hide();
    testLayout.Show();
});
}

TestWindow::~TestWindow()
{
    // Make sure that running threads terminate.
    if(Thread::GetCount())
        Thread::ShutdownThreads();
}

```

Best regards,
Oblivion
