

Hello,

I have some problems with GUI en multithreading with U++.
My app crash when a thread modify the content of a lineEdit widget.

I think an other thread is trying to read data from this lienEdit to display it at the same time.

How can i disable updating events when i modify my widgets values ?

Thank you for your help

A simple example that produce the problem.

```
#include "mt.h"

void mt::theThread()
{
    while(true) {
        String text = console.Get();
        text << "Some Text\n";
        console.Set(text);
    }
}

void mt::startButtonEvent()
{
    Thread().Run(THISBACK(theThread));
}

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

    startButton << THISBACK(startButtonEvent);
}

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

Subject: Re: Multithreading ans updating widgets
Posted by [mirek](#) on Tue, 11 Mar 2008 11:34:38 GMT
[View Forum Message](#) <> [Reply to Message](#)

cocob wrote on Tue, 11 March 2008 06:48
Hello,

I have some problems with GUI en multithreading with U++.
My app crash when a thread modify the content of a lineEdit widget.

I think an other thread is trying to read data from this lienEdit to display it at the same time.

How can i disable updating events when i modify my widgets values ?

Thank you for your help

A simple example that produce the problem.

```
#include "mt.h"

void mt::theThread()
{
    while(true) {
        String text = console.Get();
        text << "Some Text\n";
        console.Set(text);
    }
}

void mt::startButtonEvent()
{
    Thread().Run(THISBACK(theThread));
}

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

    startButton << THISBACK(startButtonEvent);
}
```

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

Only main thread can do GUI. You should use event queue and post callback that is supposed to update the widget.

Reading the widget content in a non-main thread is a bit of problem, I guess you should use shared variable that gets updated on Post event request (and is interlocked by Mutex).

Mirek
