Subject: Re: Question about PostCallback from Child Thread
Posted by kfeng on Fri, 13 Jul 2007 16:10:12 GMT
View Forum Message <> Reply to Message

luzr wrote on Fri, 13 July 2007 17:31Well, two more hints:

- you can also consider whether you should process these data to something else before posting to avoid breaking "get" and "release" library calls into two thread. If not, I think you whould have a lock for the whole library used, you can never say what is going on inside....

- also, this does not sound like a compelling case for multithreading....

According to documentation the server on the other side of the library will just continuing to queue prices until your next read.

Now that I understand MT a little bit more, you got me thinking some more. I am currently using a blocking version of the function call to grab prices, but a non-blocking one is also available.

I used to think blocking is always better, but now, I wonder if it is more efficient to poll every second using the non-blocking call in a non-MT app. I think your supposition is right - that the non-blocking will be fast enough that it shouldn't impact user experience in the GUI. A good example would be an application to view your portfolio's profit & loss - don't care if it's a second late.

However, if latency is important enough that you don't want to set a timer, like an algorithmic black-box for automated trading, then blocking + MT is probably better. But most likely greater latency will show up in other parts of the system, but since it's cumulative, it could be better to squeeze as much as you can out of it. You've opened my eyes again...

I will try both ways next week in the office...

- Ken