Subject: Question about PostCallback from Child Thread
Posted by kfeng on Wed, 11 Jul 2007 14:12:02 GMT
View Forum Message <> Reply to Message

Hi,

I have a question about PostCallback.

Let's say I call the function from a child thread (parent thread is the main GUI loop):

```
  PostCallback(callback2(myInst, &MyClass::MyCB, arg1, arg2 ));
```

Is this a blocking call (synch) or non-blocking (asynch)?
I am asking, because if it's blocking, the result should be the same as:

```
PostCallback(callback(myInst, &MyClass:MyCB2));
void
MyClass::MyCB2()
{
  arg1 = GetArg1();
  arg2 = GetArg2();
}
```

My child thread writes to arg1 and arg2 - the GUI thread reads. If PostCallback doesn't block, I can have a race condition in the second case. Asynch would also imply that PostCallback() does some kind of locking of arg1 and arg2, so the child thread can't write to it while the GUI is reading.

I'm asking because what if I need to pass a bunch (10?) of arguments. Should I put them in 1 struct/class and use callback1(). I was lazy earlier today and tried the 2nd example above, but now, I'm having second thoughts...

Can someone explain the mechanism here?
Thank you.

Regards,
Ken