

---

Subject: Need a suggestion about mouse processing inside threads

Posted by [mdelfede](#) on Mon, 12 Dec 2011 23:53:47 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Hi, I've a multi-document application, on which each document runs a threaded command loop :

```
// document command loop
void UppCadDocument::commandLoop(void)
{
    // loop not ended
    INTERLOCKED_(mutex)
    {
        loopEnded = false;
    }
    while(!Thread::IsShutdownThreads())
    {

        if(commandLine.HasCommand())
        {
            String cmd = commandLine.GetCommand();
            if(cmd != "<ESC>" && cmd != "")
                SendCommand(cmd);
        }
        Sleep(100);
    }
    // loop ended
    INTERLOCKED_(mutex)
    {
        loopEnded = true;
    }
}
```

Here, handling of commandLine (some sort of InputField) is simple; the problem arise when I've to react to mouse events; main thread can call, for example, MouseMove() on any time in the middle of anything of my document's loop.

How to synchronize it ? I can't stop my main thread waiting for a document's one is ready for event, of course... I think I'd need some sort of event-loop inside document's threads, and main thread should inject its events on it, instead executing calls directly, but how ?

This is exactly the opposite problem as the one solvable with Guilock....

My idea would be something like this :

Main thread :

```
void SomeEventHandler(somedata)
```

```
{  
    GetActiveDocument().InjectEvent(eventtype, somedata);  
}
```

Document thread :

```
while(true)  
{  
    WaitForInjectedEventsWithoutEatingCpu();  
    while(eventsInQueue)  
        ProcessPendingEvents()  
}
```

Is there something in upp that can help for all that, or I should code it using arrays of callbacks or something like that ?  
Or there's a better solution ?

Max

---