Subject: Re: MT again (SOLVED!)
Posted by mdelfede on Fri, 18 Dec 2009 11:51:12 GMT
View Forum Message <> Reply to Message

```
// calculation thread function
void CalcPage::DoCalculation(void)
{
 bool mod;

 INTERLOCKED_(mutex)  {
  inThread = true;
 }
 do
 {
  INTERLOCKED_(mutex)  {
   modified = false;
  }

  // gets the status bar
  StatusBar &sb = ((Lamell *)(GetMainWindow()))->GetStatusBar();
  {
   // signals calculation in progress
   GuiLock __;
   sb.Set("Calcolo in corso");
  }
  calcAborted = !Calculate();
  {
   // signals end of calculation
   GuiLock __;
   if(calcAborted)
    sb.Set("Errore");
   else
    sb.Set("Pronto");
  }
  INTERLOCKED_(mutex) {
   mod = modified;
  }
 }
 while(mod);
 INTERLOCKED_(mutex) {
  inThread = false;
 }
}

// modify handler -- triggers page calculation
// on keypress on every (editable) ctrl on page
void CalcPage::ModifyCb(void)
```

```
{
 bool inTh;
 INTERLOCKED_(mutex) {
  inTh = inThread;
 }
 if(!inTh)
 {
  calcThread.Run(THISBACK(DoCalculation));
  return;
 }
 INTERLOCKED_(mutex) {
  modified = true;
 }
}
```

Well, thanks Mirek !
I'll post the correct code here for reference.

The problem was that IsOpen() just checks that the thread was STARTED right, not if the thread is still active.
To check for it, I added a new variable set inside the thread (inThread).

This snippet allows to have a long time backround calculation without stopping the gui and with results displayed as soon as available.
The thread routine can handle changes in data that happens during the calculation, re-entering itself when finished.

Ciao

Max
