Subject: Re: Code before Thread.Run() nor executed
Posted by mirek on Tue, 29 Aug 2017 15:20:14 GMT
View Forum Message <> Reply to Message

rafiwui wrote on Mon, 28 August 2017 10:51How exactly does GuiLock work? Because when I tried to implement it in my application it doesn't work. Is there a documentation?
EDIT:
This is part of the code I use:

```
void MainWindow::Generate()
{
 // Generate
 GuiLock __;
 progress.progressMainLabel.SetText(t_("(0/3) Generating first..."));
 if(!GenerateFirst())
 {
  AbortGeneration(1);
  return;
 }
 progress.progressMain++;
 progress.progressMainLabel.SetText(t_("(1/3) Generating second..."));
 if(!Second())
 {
  AbortGeneration(2);
  return;
 }
 progress.progressMain++;
 progress.progressMainLabel.SetText(t_("(2/3) Generating third..."));
 if(!GenerateThird())
 {
  AbortGeneration(3);
  return;
 }
 progress.progressMain++;
 progress.progressMainLabel.SetText(t_("(3/3) Done"));
}
```

Instead of showinf every step it just generates all three and at the end updates the gui. So what do I have to change that every step it is updated?

That is because is GUI is blocked while GuiLock is active...

GuiLock is "guard class" - it locks GUI mutex in constructor and releases it in destructor. While that mutex is locked, GUI is stopped and you can meddle with variables that GUI accesses (without using PostCallback).

In above function, you lock GUI at the start of function and unlock it at the end. That is why the GUI is only updated when Generate ends...