
Subject: Re: How to update main thread UI when runing other thread

Posted by [jiuzhi](#) on Sat, 06 Feb 2010 13:04:20 GMT

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

I take a look at GuiMT. But it isn't what I want. In GuiMT, it create a task thread, But it does not need to wait for returned results

What I mean is waiting a task thread do a task and return result to main thread block;
The task thread use GuiLock to refresh UI. But under U++, when main thread waiting, the task thread call Ctrl::Refresh look does not work.

In U++, I can only find a way to do this job, create task thread, and then create a waiting thread. It is cumbersome.

I can't write code to be waiting in the main thread, otherwise the task thread can't refresh UI.

in delphi, only main thread does GUI too. But it can do this job in the main thread.

```
begin //this block run in the main thread
  create and run task thread.....
  create and run task thread.....
  Application.processmessage; //refresh UI immediately. In U++, it does not work.
  create and run task thread.....
  create and run task thread.....
  do some job.....
end
```

OR:

```
begin //this block run in the main thread
  create and run task thread.....
  create and run task thread.....
  while not result do begin //wait the task thread result and refresh UI. In U++, this will lock the
  GUI, other threads try to refresh the interface does not work
    sleep(100);
    Application.processmessage; //In delphi, this code is used to process GUI message in the main
  thread
  end;
  do some job.....
end
```

What I would like to know is the routine which does the same function in U++ like Application.Processmessage in delphi?
