Subject: Re: How to update main thread UI when runing other thread
Posted by mirek on Sat, 06 Feb 2010 10:32:06 GMT
View Forum Message <> Reply to Message

jiuzhi wrote on Sat, 06 February 2010 04:17
Thread work;
engine->working(true);
work.Run(callback(engine,&hEngine::getData));//-getData will do some UI refresh;
while (engine->working()){
  Refresh();//It cann't update UI here;
  Sleep(100);
}

I have thought of a solution is packaged into a function, and create a new Thread to run it;But it is too cumbersome.
In MFC,I can use PeekMessage;
In delphi,I can use Application.ProcessMessage;
Is there a similar solution in u++?

I am not quite sure what the problem is. However, here are several hints:

First, in U++, only main thread does GUI.

PostCallback - any thread can post a callback that gets executed in the main thread

GuiLock - any thread can access GUI objects directly

Ctrl::Call - you can even "call" routines in the main thread. This is done by placing callback to the queue and waiting until it gets processed by the main thread. In fact, most calls to Ctrl routines, like performing message loop, automatically redirect to the main thread using this Call (or similiar ICall).

Please check these examples:

reference/GuiLock
reference/GuiMT