Subject: Re: Select Grid Row BY ID
Posted by Oblivion on Wed, 10 May 2017 17:09:26 GMT
View Forum Message <> Reply to Message

Hello germax,

Quote:The problem (the reason why I went with multithreading in the first place) is still persistent. the GUI blanks out and the App is irresponsive for quite some time (until all workes are finished)

Do you really need CoWorker? I don't know the details of your app, but the example you provided is somewhat complicated, imo.
Spawning CoWorkers per row, as your code seems to do, would be an overkill with a considerable overhead.
If all you need is an asynchronous (non-blocking) gui, you can easily use a single worker thread (Thread, not CoWorker) and get rid of the additional vector.
Please find attached a simple example with a csv file over 6000 lines. It simply removes entries one by one without blocking the gui. And it also demonstrates how to halt a thread.
It'll give you the idea, I hope. :)

The thread code will look like:

```
void MTT::processFile()
{
 int rowc = grid.GetRowCount();
 progin.Percent();

        // Below is the actual thread function. We are taking advantage of C++11 lambdas here.
 Thread().Run([=] {
  for(auto i = rowc - 1, j = 0; i >= 0; i--, j++) {
   if(IsShutdownThreads())
    break;
//   int q = grid(i, 2);
//   if(q <= 0)
//    this->RemoveItem(i, j);  // You can also use a dedicated method.
    PostCallback([=] {
     grid.Remove(i);
     progin.Set(j, rowc - 1);
    });

   Sleep(10);
  }
 });
}
```

Actually you may even not need multithreading at all.
if you are using a for/while loop in single threaded environment which thakes time and thus blocks your app, I mean if that's the only problem, you can use Ctrl::ProcessEvents() to refresh the GUI inside your loop.

E.g.

```
for(auto i : very_large_vector_to_process) {

 DoYourStuff();
 ProcessEvents();
}
```

Regards,

Oblivion

File Attachments

1) MTtest.zip, downloaded 352 times