
Subject: Re: Select Grid Row BY ID
Posted by [Oblivion](#) on Thu, 11 May 2017 18:04:35 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello germax, and you're welcome.
Ultimate++ forums are full of helpful people.

Quote:Your example is just what I wanted.. it doesn't explain to me why mine didn't work, but all I care about now is that it DOES work one way or another Very Happy so I'm good.

Sorry I forgot to write that, The reason is actually pretty simple. You use `CoWork::Finish()`, which is blocking.

You should have used `CoWork::IsFinished()` instead.

In that case, all you need to call is `Ctrl::ProcessEvents()`. But still, `CoWork` seems to me an overkill here.

All you need to do is to alter your code as below (Of course you don't need to use lambda functions and while in the example accessing the vector from threads does not pose classic concurrency problems, you should consider using mutex (`CoWork::Finlock()` will do fine. See docs and tutorials for more explanation.):

```
void MTT::processFile()
{
    int rowc = grid.GetRowCount();
    Vector<bool> del;
    del.SetCount(rowc, false);
    progin.SetTotal(rowc);
    progin.Percent();

    for(int i=0; i < rowc; i++)
        cw & [=, &del] {
            // Now, this isn't the proper way to accaess shared elements (UI or core) from a
            // thread. But ONLY in this example, it won't do harm.
            grid.GetRow(i);
            int q = grid(i,2);
            if(q <= 0)
                // CoWork::FinLock();
                del.Set(i, true);
        };
    // cw.Finiehed()
    // Non blocking way:
    while(!cw.IsFinished())
        ProcessEvents();

    for(int i = rowc-1; i>=0; i--)
    {
        progin++;
        if(del[i]) grid.Remove(i);
    }
}
```

```
ProcessEvents();  
}  
RDUMP(del);  
}
```

This is all you need to do.

Regards.

Oblivion.
