
Subject: Re: Select Grid Row BY ID

Posted by [germax](#) on Mon, 15 May 2017 12:22:16 GMT

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

yeah.. not an issue really..

tried both (as you can see in true multithreading I use a vector to store which line gets to be removed from within main thread...

And I've tested that with a single worker as well.

I think the main issue with oblivions code not working properly on my machine is that sql connections are not exactly thread safe.

that's why it fails unpredictably at any of the first say twenty rows.

(sometimes on row 2, sometimes at row 12.. but always at an SQL Insert or Update, not on an SQL Select for some unknown reason)

PerThread() is making things worse on my single core for some reason.

(maybe sqlite doesn't like multithreading.. haven't checked *shrugs*)

Anyways, there are many things that do not work well at all;

some are even perplexingly odd in terms of "why you no work you punk"

Take the ProgressInfo for example:

ProgressInfo prog in header (as per oblivions example)

prog++ (as per the ProgressInfo A++ is failing claiming operator ++ not available)

prog.operator++() (that looks sooooo wrong) is working flawlessly...

Ah well, you win some you loose some

Oh btw.. as soon as there is no sql involved at all.. oblivions code is woking just nicely,

(with line removal) it even works with "forward iteration" removing lines and accessing the same line again (since it's the next unprocessed one)

manipulating the grid either way in a thread is troublesome..

adding a column is rather pointless imho if that information can be stored elsewhere

for the second it'll be usefull. (gridctrl is quite complex in terms of storage... a vector is of much smaller footprint)

passing the row as a raw vector to the function has also been tried..

no improvement at all (but a new container thus more memory in use)

it's always the SQL causing troubles.

What I really "needed" has been resolved by oblivion already (non locking gui) in a single threaded main (no workers whatsoever)

Apart from that, visual representation (coloring etc)

itself is more eye candy than of certain use while the processing is in progress.

but I'd have to mark rows (change background colors) afterwards anyways so why not directly saves me having another loop to do just that

(and frankly it's the exact same thing as changing a grid fields value, right)

