

## Wt - Bug #5559

### Server push - concurrency issue

02/22/2017 09:24 AM - Max Quatember

<b>Status:</b>	Feedback	<b>Start date:</b>	02/22/2017
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Roel Standaert	<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>			
<b>Description</b>			
Hi!			
We have a concurrency problem with UI-locks during server push operations:			
We have a spawned thread that shows a modal dialog with a progressbar, builds our data model and pushes progressbar updates to the client UI.			
It's quite similar to the example in your ServerPush.C			
During this thread it is possible that another request for the same application becomes active, gains the UI-lock and modifies the data model.			
This leads to a corrupted heap sometimes, when the two threads access the same data members.			
Is there a possibility to get the UI-lock (before the auxiliary thread) and push changes without quitting the UI-lock?			
I can extract a testcase showing our problem but this would take some time, maybe there is some quick help or hint from your side...			
Best regards,			
Max			

### History

#### #1 - 03/01/2017 09:35 AM - Roel Standaert

- Status changed from New to Feedback

- Assignee set to Roel Standaert

I'm not sure I understand the situation completely. Whenever you want to change anything to the widget tree of a WApplication, or anything associated with it, like item models, you should take the update lock first.

Are you trying to change things without taking the update lock, or are you experiencing concurrency problems even when you have taken the update lock?

Regards,

Roel

#### #2 - 03/01/2017 10:55 AM - Max Quatember

Hi Roel!

My concurrency problem is:

During a server push in an auxiliary thread, between two UI locks, a new request for the same session on the main thread can become active and take the UI lock.

I would need to lock the UI for the whole time of a server-push and push multiple changes during one lock.

Is this clear enough?

Best regards,

Max

**#3 - 03/02/2017 03:43 PM - Roel Standaert**

There's no way to do this in Wt currently. You'd have to implement your own additional locking, or use a [modal dialog](#) to block events.

**#4 - 03/02/2017 04:31 PM - Max Quatember**

Hi Roel!

OK, I understand.

Our dialog that shows the progress is not modal. It is "show"n and not "exec"uted so it does not block the current request.

But If I "exec" the dialog, where is the place to start the auxiliary thread to server-push the progress?

We already tried to implement an additional locking, without success.

For me there are 2 ways to go on for now:

1) There is a way - that I don't know - to server-push changes from an auxiliary thread to a modal dialog.

2) You can implement us a way to do this in Wt.

Maybe we can also talk to Koen, he has a deeper insight in our application and history with Wt.

Best regards,

Max

**#5 - 03/02/2017 06:54 PM - Koen Deforche**

Hey max,

Modal and using exec() is not the same. A modal dialog simply prevents other events from being generated and handled until the dialog is closed. Exec() starts a recursive event loop (blocking the current thread) until the dialog is closed.

You can also have a modal dialog and show() it.

Regards,

Koen

**#6 - 03/03/2017 09:28 AM - Max Quatember**

Hi Koen!

Please excuse me, I confused myself.

We do "show" a "modal" dialog. Nevertheless we can force another event to get generated...

Best regards,

Max

**#7 - 03/03/2017 10:12 AM - Koen Deforche**

Hey Max,

You mean an event related to a widget from outside the dialog is being delivered? This shouldn't happen, it is actively being prevented within the session event handling. If so, then that's a bug.

Regards,

koen

**#8 - 03/03/2017 12:30 PM - Max Quatember**

Hi Koen!

Yes thats what I mean.

Post-mortem dumps from our customers show this situation.

We can reproduce the situation in about 1 of 100 cases.

For us its a minor but annoying problem - Beside the great performance and stability Wt delivers!

Our solution would be to globally get the UI-lock and push multiple updates during this UI-lock.

But this requires your help.

Best regards,

Max

**#9 - 03/04/2017 12:30 AM - Koen Deforche**

- *Tracker changed from Support to Bug*

Hey,

Can you provide some more detail of the direction to look into? Does the port-mortem indicate there is some kind of race condition in showing the dialog and having the signal, or something else?

Koen

**#10 - 03/13/2017 12:06 PM - Max Quatember**

- *File WebUI\_WarmupAndConcurrentDraw.txt added*

Hi Koen!

I have a stacktrace of one of our crashes.

Thread 1:

Is the spawned thread after the modal dialog is shown.

Thread 2:

Is the concurrent request.

Is this enough information for you?

Max

**#11 - 01/16/2018 02:41 PM - Max Quatember**

Hi!

This can be closed, worked around this issue!

Best regards,

Max

**Files**

---

WebUI_WarmupAndConcurrentDraw.txt	26.2 KB	03/13/2017	Max Quatember
-----------------------------------	---------	------------	---------------