

## Wt - Bug #976

### segmentation fault when creating a WMessageBox when not executed on a session thread

09/04/2011 01:40 PM - Arjan Vermeij

<b>Status:</b>	Closed	<b>Start date:</b>	09/04/2011
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Koen Deforche	<b>% Done:</b>	0%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	3.3.4		

#### Description

I tried an example from the documentation:

```
const Wt::StandardButton result = Wt::WMessageBox::show ("Confirm", ("About to wreak havoc... Continue ?"), (Wt::Ok | Wt::Cancel));
```

If not executed on a session thread, this yields a segmentation fault, gdb says:

```
Program received signal SIGSEGV, Segmentation fault.
[Switching to Thread 0x7fffe4ff9700 (LWP 12041)]
Wt::WebRequest::getParameterValues (this=0x0, name=...) at /home/arjan/bzr/casw/moos-ivp/external-software/wt/src/web/WebRequest.C:101
101  Http::ParameterMap::const_iterator i = parameters_.find(name);
(gdb) bt
#0  Wt::WebRequest::getParameterValues (this=0x0, name=...) at /home/arjan/bzr/casw/moos-ivp/external-software/wt/src/web/WebRequest.C:101
#1  0x00007ffff79f7299 in Wt::WebRequest::getParameter (this=0x0, name=...) at /home/arjan/bzr/casw/moos-ivp/external-software/wt/src/web/WebRequest.C:93
#2  0x00007ffff7a06191 in Wt::WebSession::notify (this=0x7fffe8003120, event=...) at /home/arjan/bzr/casw/moos-ivp/external-software/wt/src/web/WebSession.C:1561
#3  0x00007ffff7a07da5 in Wt::WebSession::doRecursiveEventLoop (this=0x7fffe8003120) at /home/arjan/bzr/casw/moos-ivp/external-software/wt/src/web/WebSession.C:901
#4  0x00007ffff7846334 in Wt::WDialog::exec (this=0x7fffe4ff73c0, animation=<value optimized out>) at /home/arjan/bzr/casw/moos-ivp/external-software/wt/src/Wt/WDialog.C:228
```

#### History

##### #1 - 09/12/2011 10:44 AM - Koen Deforche

- Assignee set to Koen Deforche

Hey Arjan,

I've tried some scenarios for this and indeed it fails. I do not readily see a single problem that causes this, it might be more than one :-)

How are you doing this actually, using server push with UpdateLock or with WServer::post() ?

Regards,

koen

##### #2 - 09/12/2011 10:45 AM - Koen Deforche

- Status changed from New to InProgress

##### #3 - 07/01/2013 06:51 PM - Pierluigi Vicinanza

Hi,

I think I have a several repro for this bug:

- 1) create a custom WDialog dialog (e.g. MyDialog), with a WFileUpload widget inside
- 2) connect WFileUpload::onFileTooLarge signal to a MyDialog method (e.g. MyDialog::onLargeFileUploaded())
- 3) make "MyDialog::onLargeFileUploaded()" display an error message by calling "Wt::WMessageBox::show("Error", "file too large");"

4) start you wt app and make it start a **blocking** MyDialog by calling MyDialog::exec()

5) upload a large file using the WFileUpload widget provided by MyDialog

6) wait for upload to finish and error message to be displayed.... BOOM!!

Alternatively, you can use these steps:

1) create a custom WDialog dialog (e.g. MyDialog)

2) create a method "MyDialog::displayError()" that displays an error message by calling "Wt::WMessageBox::show("Error", "Blah!");

4) start you wt app and make a session execute a **blocking** MyDialog by calling MyDialog::exec()

5) from your main app, post an event (e.g. via WServer::post()) to your wt session started in step 5, this function will have to execute the "MyDialog::displayError()" on

6) wait for event to be notified and error message to be displayed.... BOOM!!

Key elements to trigger this bug are:

a) having a **blocking** dialog running

b) starting another **blocking** dialog (WMessageBox::show) on top of the previous one, but **from a different session**

I've done a (very brief) debugging session and the problem seems to be one of the two recursive event loops that is clearing the connection handler after having "consumed" it and while the other event loop is still referencing it.

Now, considering what Wt docs say about the topic (ref. [http://www.webtoolkit.eu/wt/doc/reference/html/classWt\\_1\\_1WDialog.html](http://www.webtoolkit.eu/wt/doc/reference/html/classWt_1_1WDialog.html)):

"A thread that is locked by a recursive event loop cannot be used to process requests from another sessions",

and that blocking/synchronous dialogs are somehow deprecated, I'm not sure this can still be considered a "valid" bug (assuming that the "incorrect usage" explained above is the only one that it's able to trigger it).

Hope this helps.

Regards,

Pierluigi Vicinanza

#### #4 - 10/22/2014 04:19 PM - Koen Deforche

- Status changed from *InProgress* to *Resolved*

- Target version set to 3.3.4

Hey,

This is indeed not supported and would be hard to implement. Given the workaround of not using blocking APIs all-together, we've updated the documentation to indicate this limitation.

Koenr

#### #5 - 03/17/2015 08:25 AM - Koen Deforche

- Status changed from *Resolved* to *Closed*