

Wt - Bug #9363

Add a workaround for Boost Spirit issue #688

11/23/2021 03:00 PM - Roel Standaert

Status:	Closed	Start date:	11/23/2021
Priority:	Normal	Due date:	
Assignee:	Roel Standaert	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	4.5.2		
Description			
Since Boost 1.76.0 there is a regression in the outputting of numbers: https://github.com/boostorg/spirit/issues/688			
Wt depends on this to serialize doubles to JavaScript.			
Related issues:			
Related to Bug #9572: Some JSON serializer tests fail with recent Boost versions		Resolved	01/05/2022

History

#1 - 11/23/2021 04:19 PM - Roel Standaert

- Status changed from *InProgress* to *Review*

#2 - 11/30/2021 01:30 PM - Roel Standaert

- Status changed from *Review* to *Implemented @Emweb*

#3 - 12/12/2021 03:59 PM - Roel Standaert

- % Done changed from 0 to 100

#4 - 12/12/2021 04:00 PM - Roel Standaert

- Status changed from *Implemented @Emweb* to *Resolved*

#5 - 12/13/2021 01:08 PM - Bruce Toll

Thanks for all of the work with upstream on this. I was wondering if the issue could also impact `double_to_s` and `real_to_s` in `src/Wt/Dbo/backend/Postgres.C`. I haven't had the opportunity to test with Boost 1.76+ yet.

#6 - 12/14/2021 09:50 AM - Roel Standaert

It may affect those functions as well, indeed.

#7 - 12/14/2021 10:43 AM - Roel Standaert

The bug occurs when it has to round the input up in order to fit into the given precision, so in our `round_js_str` we get a double as input, but ask it to output it with at most 7 decimals.

Our `real_to_s` function on the other hand gets a float as input, and thus already has less precision, so at the very least I can't reproduce the issue with the numbers I used to test `round_js_str`. That suggests we may also be able to work around the issue by sacrificing precision beforehand, or increasing precision where we use `round_js_str`.

If the bug affects `Postgres.C` as well, we're a lot less likely to trigger it. I will look further to see if there is any number that will trigger the bug.

#8 - 12/14/2021 11:37 AM - Roel Standaert

I tried to start from 9.999f and using `nextafterf` to near 10.f, and didn't notice the bug.

I did encounter another issue where `stof` would throw an exception, though.

#9 - 01/05/2022 11:51 AM - Roel Standaert

- Related to Bug #9572: Some JSON serializer tests fail with recent Boost versions added

#10 - 01/05/2022 11:51 AM - Roel Standaert

- Status changed from *Resolved* to *Closed*