

Wt - Bug #9596

Possible regression with Wt Dbo floating-point precision (Postgres backend)

01/10/2022 01:16 PM - Bruce Toll

Status:	InProgress	Start date:	01/10/2022
Priority:	Normal	Due date:	
Assignee:	Roel Standaert	% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:	4.10.0		

Description

It appears that there may be a small loss of precision with some floating-point values retrieved from Postgres when using Wt Dbo 4.6.1, compared with 4.5.1.

The attached file 0001-Add-regression-tests-for-floating-point-retrieval.patch adds two tests that illustrate the issue.

Applying the above patch to Wt 4.6.1 and running test.postgres results in 12 errors in my environment (gcc 11.2.0, boost 1.74.0, postgres 12.9):

```
$ ./test.postgres -l message -t '*/dbo_test48, dbo_test49' -e stdout 2>/dev/null
Running 2 test cases...
DboTest.C(3422): error: in "dbo_test48": check in_value.as_float == out_value.as_float has failed
[3.35419397e-37 != 3.35419419e-37]
DboTest.C(3422): error: in "dbo_test48": check in_value.as_float == out_value.as_float has failed
[0.0001000000005 != 0.0001000000012]
DboTest.C(3422): error: in "dbo_test48": check in_value.as_float == out_value.as_float has failed
[0.000106811516 != 0.000106811523]
DboTest.C(3422): error: in "dbo_test48": check in_value.as_float == out_value.as_float has failed
[0.98999995 != 0.98999989]
DboTest.C(3422): error: in "dbo_test48": check in_value.as_float == out_value.as_float has failed
[2.71828175 != 2.71828151]
DboTest.C(3464): error: in "dbo_test49": check in_value.as_double == out_value.as_double has failed
[9.999999999999991e-05 != 0.0001]
DboTest.C(3464): error: in "dbo_test49": check in_value.as_double == out_value.as_double has failed
[0.9999999999999989 != 1]
DboTest.C(3464): error: in "dbo_test49": check in_value.as_double == out_value.as_double has failed
[1.0000000000000007 != 1.0000000000000009]
DboTest.C(3464): error: in "dbo_test49": check in_value.as_double == out_value.as_double has failed
[1000.0000000000001 != 1000]
DboTest.C(3464): error: in "dbo_test49": check in_value.as_double == out_value.as_double has failed
[99999999.999999985 != 100000000]
DboTest.C(3464): error: in "dbo_test49": check in_value.as_double == out_value.as_double has failed
[9.999999999999987e+17 != 1e+18]
DboTest.C(3464): error: in "dbo_test49": check in_value.as_double == out_value.as_double has failed
[1.0000000000000001e+18 != 1e+18]
```

NOTE: File paths and test names shortened for clarity.

These errors may be due to commit 81ec366f that replaced std::stof and std::stod with a parser based on Boost Spirit. There is an upstream issue with the Boost Spirit library that seems related and is still unresolved: <https://github.com/boostorg/spirit/issues/668>.

Repeating the test with test.sqlite3, no errors are reported.

Also, restoring the 4.5.1 version of src/Wt/Dbo/backend/Postgres.C (with all other source files from 4.6.1) and repeating the tests, both test.postgres and test.sqlite3 completed without error. However, the subnormal tests in dbo_test45 failed due to the omission of the subnormal fix in commit 81ec366f (added after 4.5.1, see issue [#9490](#)).

For your review, I have attached a patch that attempts to address the precision regression, while retaining the subnormal fix. The patch restores the 4.5.1 stof/stod implementation while adding exception handling to retry std::out_of_range errors with the Boost Spirit code (to handle subnormals). I haven't checked how much overhead this exception handling adds, but the updated code passes the subnormal tests (dbo_test45), as well as the new floating-point retrieval tests (dbo_test48 and dbo_test49).

See related: [#9595](#). If I can provide any additional information, please let me know.

History

#1 - 01/10/2022 06:58 PM - Roel Standaert

- Target version set to 4.6.2

We'll take a look at this. I was considering the use of `to_chars/from_chars` where available instead of relying on Boost Spirit.

In the long run I would prefer to banish any locale-dependent functions, like `std::stof` and `std::stod`, from the Wt source code, so that Wt continues to work properly regardless of global locale.

#2 - 01/10/2022 11:10 PM - Bruce Toll

Thanks,Roel!

#3 - 01/14/2022 12:05 PM - Bruce Toll

A few additional notes on the attached file, 0001-Add-regression-tests-for-floating-point-retrieval.patch:

1. In `dbo_test48`, I used the syntax `"CAST(value as FLOAT(24))"` in an effort to reduce postgres-specific syntax/types and possibly make the test usable with other backends. Likewise with the `"CAST(value as FLOAT(53))"` in `dbo_test49`.
2. I probably should have used single quotes around the value in the above casts, e.g. `"CAST('value' as FLOAT(24))"`. For Postgres, I do not believe it makes a difference in the test results. The value is initially parsed as a numeric and then cast. My focus was on the potential regression in Postgres.C. But if the tests are included in Wt or used with other databases, the single quotes should probably be added. Of course, there may be other portability issues...
3. The test values were selected using the configuration described in [#9595](#) with the two attached patch files applied. As described there, the tests should pass with two warnings on subnormal values. I then partially reverted the second patch to use the original Wt 4.6.1 `getResult` code in Postgres.C for loading, while retaining the updated `float_to_s` and `double_to_s` (for better precision when saving). I then ran the tests with 'message' level logging and selected some of the "PG text value:" values for cases that did not round-trip.

Thanks, again, for your reply above and taking the time to review this issue.

#4 - 02/07/2022 05:06 PM - Roel Standaert

- Status changed from New to InProgress

#5 - 02/07/2022 05:06 PM - Roel Standaert

- Assignee set to Roel Standaert

#6 - 02/14/2022 12:47 PM - Roel Standaert

- Target version changed from 4.6.2 to 4.7.0

#7 - 03/10/2022 03:23 PM - Roel Standaert

- Target version changed from 4.7.0 to 4.8.0

#8 - 07/06/2022 11:49 AM - Roel Standaert

- Target version changed from 4.8.0 to 4.9.0

#9 - 10/06/2022 09:52 AM - Roel Standaert

- Target version changed from 4.9.0 to 4.10.0

Files

0001-Add-regression-tests-for-floating-point-retrieval.patch	3.51 KB	01/10/2022	Bruce Toll
0002-Prefer-stof-stod-over-boost-to-retain-precision.patch	2.12 KB	01/10/2022	Bruce Toll