Wt - Bug #9039

Wt::WDateTime::toLocalDateTime not working

09/03/2021 03:57 PM - Torsten Schulz

Status:	New	**Start date:**	09/03/2021
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

Following code crashes:

```
            auto test = Wt::WDateTime::currentDateTime();
            auto locale = Wt::WLocale("de_DE");
            auto localTest = test.toLocalTime(locale);
            std::cout << test.toString() << std::endl;
            std::cout << localTest.toString() << std::endl;
```

in last line.

History

#1 - 09/07/2021 10:27 AM - Korneel Dumon

Hi,

you are right, the reason is that the WLocale does not have any timezone information configured. This should be done manually using WLocale::setTimeZone()
We have a fix pending for the 4.5.1 release (coming soon :)) that will log a warning and throw an exception instead of crashing.

Related to 8039.

#2 - 09/08/2021 08:57 AM - Torsten Schulz

Korneel Dumon wrote in #note-1:

> Hi,
>
> you are right, the reason is that the WLocale does not have any timezone information configured. This should be done manually using WLocale::setTimeZone()
> We have a fix pending for the 4.5.1 release (coming soon :)) that will log a warning and throw an exception instead of crashing.
>
> Related to 8039.

Ok, that's good. But then I have another Problem. When I try to use date::locate_zone, the application crashs. I think it is because I use a different version of the date library, so it would be helpful when you'd add that function to WEnvironment.

When I try date::locate_zone without witty, it works. When I don't link date-tz, I get a compile error.

#3 - 09/08/2021 07:37 PM - Torsten Schulz

I spend a lot time into it last days. So I found out that the example "locale" is compiling and running.
But when I add in an own app and use date::locate_zone("..."), then I get a compiler error:

main.cpp:(.text+0x30): undefined reference to `date::locate_zone(std::basic_string_view >)'

I think that I have to add something, but I don't find out what I need. I added the libwt.so, which should be the right lib as I had seen by inspecting code and libwt.so - but it doesn't work.

#4 - 09/08/2021 10:40 PM - Torsten Schulz

I tried with that:

main.cpp:

```
#include <Wt/Date/tz.h>
#include <iostream>
```

```
int main(int argc, char **argv) {
        try {
            auto zone = date::locate_zone("Europe/Berlin");
            std::cout << __LINE__ << std::endl;
            std::cout << zone->name() << std::endl;
        } catch (const std::exception &e) {
            std::cout << e.what() << std::endl;
        }
}
```

CMakeLists.txt:

```
cmake_minimum_required(VERSION 3.8)

set(CMAKE_INCLUDE_CURRENT_DIR ON)

set(CMAKE_CXX_STANDARD 20)
set(CMAKE_CXX_STANDARD_REQUIRED ON)

project(tztest.wt VERSION 2.0.0)

set(CMAKE_CXX_COMPILER "/usr/bin/g++-11")
set(CMAKE_INCLUDE_CURRENT_DIR ON)

add_executable(${PROJECT_NAME}
    main.cpp
)
target_link_libraries(${PROJECT_NAME}
    wt
    wthttp
)
```

Error:

```
/usr/lib64/gcc/x86_64-suse-linux/11/../../../../x86_64-suse-linux/bin/ld: CMakeFiles/tztest.wt.dir/main.cpp.o:
 in function `main':
main.cpp:(.text+0x30): undefined reference to `date::locate_zone(std::basic_string_view<char, std::char_traits
<char> >)'
collect2: error: ld returned 1 exit status
make[2]: *** [CMakeFiles/tztest.wt.dir/build.make:97: tztest.wt] Fehler 1
make[1]: *** [CMakeFiles/Makefile2:83: CMakeFiles/tztest.wt.dir/all] Fehler 2
make: *** [Makefile:91: all] Fehler 2
```

#5 - 09/09/2021 09:31 AM - Korneel Dumon

I suspect it has something to do with the fact that you are using c++20. I'm not really sure how far along we are in the whole "standalone vs std date library" story. My colleague who worked on this is on holiday this week, but I will ask him to follow up on this when he is back.

#6 - 09/09/2021 02:05 PM - Torsten Schulz

Ah, sorry. I forgot to mention. I also thought that could be the reason. But I also tried with g++-10 and c++17 - same result.

#7 - 09/15/2021 03:15 PM - Roel Standaert

This is because of a C++ standard mismatch. Wt by default is built in C++14 mode, where string_view is not available. However, if you then use Wt in C++17 or higher mode then it will look for the missing string_view. You'll have to make sure that if you're using Wt with C++17, that Wt itself is also compiled with CMAKE_CXX_STANDARD set to 17.

#8 - 09/15/2021 03:16 PM - Roel Standaert

I'm not sure if there's much we can do to improve this, it's just a caveat. We try not to modify the date library.

#9 - 09/15/2021 05:42 PM - Torsten Schulz

Ok, I understand it. But maybe I'm not able to use cmake correct.

I tried cmake -DCMAKE_CXX_STANDARD=20 ../

But still the same problem.

#10 - 09/17/2021 12:13 PM - Torsten Schulz

Ok, me again. I got it compiling when I remove the build directory and make it new.

The link error is indeed not existing any longer then.

But I don't know if the bug can be marked as fixed, because then the library isn't working any longer. Remove a widget is freezing the app then. It doesn't matter if the remove is manually by myself or automatically (i.e. rebind in a template).