

Wt - Bug #7719

Convert a cpp17::any with unsigned long to a string always throws.

09/17/2020 11:51 AM - Andreas Frolov

Status:	Resolved	Start date:	09/17/2020
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:			
Description			
The following code always throws:			
<pre>Wt::cpp17::any value = 4ul; auto s = Wt::asString(value);</pre>			
because you try to cast value to the type long instead of unsigned long what leads to an exception.			

History

#1 - 09/25/2020 01:47 PM - Roel Standaert

What platform is this on? It's working fine on Linux 64 bit, but maybe the issue is specific to 32 bit, I'll have to check that.

#2 - 09/25/2020 03:10 PM - Andreas Frolov

I tested on Windows 32 bit, but I wonder that it works somewhere.

This is your code:

```
@ } else if (v.type() typeid(unsigned long)) {
if (sizeof(long) == 4) {

if (format.empty())

return WLocale::currentLocale().toString

((unsigned)cpp17::any_cast(v));

else {

char buf[100];

snprintf(buf, 100, format.toUTF8().c_str(),

(unsigned)cpp17::any_cast(v));

return WString::fromUTF8(buf);

}

} else {

if (format.empty())

return WLocale::currentLocale()

.toString((::uint64_t)cpp17::any_cast(v));

else {

char buf[100];

snprintf(buf, 100, format.toUTF8().c_str(),

(::uint64_t)cpp17::any_cast(v));

return WString::fromUTF8(buf);
```

```
}  
}  
}@
```

All cases in this block try to execute any_cast with wrong type.

#3 - 09/25/2020 03:13 PM - Roel Standaert

I just tested on Windows and found the issue myself. I'm fixing it. I think on Linux 64 bit it takes a different code path since there unsigned long is synonymous with uint64_t.

#4 - 09/25/2020 03:23 PM - Roel Standaert

- Status changed from New to Resolved

I pushed a fix

#5 - 10/01/2020 11:19 AM - Andreas Frolov

You only fixed 2 out of 4 cases. I think an exception is still thrown on OSX (else case).

<https://stackoverflow.com/questions/36814040/osx-vs-linux-how-to-deal-with-unsigned-long-and-uint64-t>

#6 - 10/01/2020 11:35 AM - Roel Standaert

I see.