
Subject: Re: Improved std::[w]string support
Posted by [Klugier](#) on Wed, 03 Feb 2021 19:07:19 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello Mirek,

It seems that this change cause compilation problem on Linux with GCC (a lot of warnings):

```
/home/klugier/upp/uppsrc/Core/Value.h: In function 'bool Upp::operator!=(Upp::String, const
Upp::Value&)':
/home/klugier/upp/uppsrc/Core/Value.h:341:73: error: call of overloaded 'String(const
Upp::Value&)' is ambiguous
 341 | inline bool operator!=(T x, const Value& v) { return v.Is<VT>() ? (VT)v != x : v != Value(x);
} \
    |                                     ^
/home/klugier/upp/uppsrc/Core/Value.h:343:26: note: in expansion of macro
'VALUE_COMPARE_V'
 343 | #define VALUE_COMPARE(T) VALUE_COMPARE_V(T, T)
    |             ^~~~~~
/home/klugier/upp/uppsrc/Core/Value.h:351:1: note: in expansion of macro 'VALUE_COMPARE'
 351 | VALUE_COMPARE(String)
    | ^~~~~~
In file included from /home/klugier/upp/uppsrc/Core/Core.h:302,
               from /home/klugier/upp/uppsrc/ide/Builders/BuilderUtils.h:4,
               from /home/klugier/upp/uppsrc/ide/Builders/BuilderUtils.cpp:1:
/home/klugier/upp/uppsrc/Core/String.h:403:2: note: candidate: 'Upp::String::String(const string&)'
 403 | String(const std::string& s) { String0::Set0(s.c_str(), (int)s.length()); }
    | ^~~~~~
/home/klugier/upp/uppsrc/Core/String.h:386:2: note: candidate:
'Upp::String::String(Upp::String&&)'
 386 | String(String&& s) { String0::Pick0(pick(s)); }
    | ^~~~~~
/home/klugier/upp/uppsrc/Core/String.h:385:2: note: candidate: 'Upp::String::String(const
Upp::String&)'
 385 | String(const String& s) { String0::Set0(s); }
    | ^~~~~~
```

Klugier
