

# C and C++ Compatibility Study Group

## Meeting Minutes (May 2022)

Reply-to: Aaron Ballman (aaron@aaronballman.com)

Document No: N2987

SG Meeting Date: 2022-05-06

Fri May 06, 2022 at 11:00am EST

### Attendees

Aaron Ballman	WG21/WG14	chair
Hubert Tong	WG21/(14)	scribe
Michael Wong	WG21/(14)	
Jens Gustedt	WG14	
Steve Downey	WG21	
Will Wray	(14)/(21)	
Philipp K. Krause	WG14	

Code of Conduct: follows ISO, IEC, and WG21 CoCs (no current WG14-specific CoC)

### Agenda

WG14 (<http://www.open-std.org/jtc1/sc22/wg14/www/docs/n2888.htm>) N2888 Require exact-width integer type interfaces

WG14 (<http://www.open-std.org/jtc1/sc22/wg14/www/docs/n2889.htm>) N2889 Pointers and integer types

### WG14 N2888 Require exact-width integer type interfaces

Jens G presents.

Hubert raises that this still requires changes to implementations before they can consider `__int128_t` as an extended integer type: `printf`, etc. and either literal suffixes or allow unsuffixed literals with values requiring such a width (for `[U]INT_C` to work)

Jens G agrees with that assessment.

Hubert notes the current paper does not require that `__int128_t` be an extended integer type but is concerned that other papers would lead to that effect.

Aaron asks to allow implementations to behave unpredictably if, for example, `INTMAX_C` is used with a value that exceeds the range of `intmax_t`.

Hubert notes that is already the case: 7.20.4 p2

Aaron raises potential implementation concern over library/compiler split on support for the types.

Hubert points out that some literals with values requiring such a width are currently allowed by some implementations (GCC) with a warning but with different semantics (truncates).

## WG14 N2889 Pointers and integer types

Hubert raises that roundtrip value conversion does not work for IBM i because the set of pointer-validity-preserving operations are restricted.

[ various questions ]

Jens G concerned about programmers being affected by such restrictions.

Hubert notes that there is some general sentiment that the platform is already viewed as requiring special treatment when porting.

Hubert raises concern that the definition of `[u]intptr_t` means that 128-bit pointer platforms would need to do the extra work to provide 128-bit integer types with all the bells and whistles from N2888 and other papers (not just status quo `__[u]int128_t` compiler support).

Hubert suggests that the `[u]intptr_t` definition be changed to allow a reduced set of requirements.

Jens G replies that he thinks that would be difficult.

Steve asks if this paper will cause a need for an `atomic_[u]intptr_t`

Aaron discovered that those were already required (oops!)

Aaron notes that N2960 (`_BitInt` fixes) will address further concerns over whether `[u]intptr_t` can be bit-precise types.

Hubert gets clarification from Aaron that `_BitInt` changes are not reflected in the latest Working Draft, so the concerns mentioned are not apparent if reviewing the current paper without extra `_BitInt` context.

## Wrapup

Aaron: I'll schedule the meeting for June shortly. I'm a bit worried about the quorum problems we've had since switching the meeting time, I may consider switching back to our old time.

End at 12:17pm EST