C and C++ Compatibility Study Group Meeting Minutes (Jul 2021)

Reply-to: Aaron Ballman (aaron@aaronballman.com)
Document No: N2789
SG Meeting Date: 2021-07-16

Fri Jun 16, 2021 at 1:00pm EST

Attendees

Aaron Ballman	WG14 WG21 chair
Jens Maurer	WG21
Clive Pygott	WG14 (21)
Jens Gustedt	WG14 (21)
Will Wray	(14) (21)
Martin Uecker	WG14
Robert Seacord	WG14 scribe
Ville Voutilainen	WG21 WG14
Tom Honnermann	WG21
Gauthier Harnisch	
Mark Zeren	WG21
Philipp Klaus Krause	WG14
Ronan Keryell	WG21
Patrice Roy	WG21
Corentin Jabot	WG21

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

Agenda

Discussing the following papers:

WG21 P2290R0 (https://wg21.link/p2290R0) Delimited escapes sequences
WG21 P2316R0 (https://wg21.link/p2316R0) Consistent character literal encoding
WG21 P2295R4 (https://wg21.link/p2295R4) Support for UTF-8 as a portable source file encoding

P2290R0 Delimited escapes sequences

Corentin Jabot presenting
Consistent with P2071 to get name using \n
Not used by any C++ compiler that we are aware of.
Don't know about C compilers.
This paper was tentatively approved by evolution in C++ with the caveat that.
The C committee is not affected by the feature so that C++ can adopt without concern.
Would be a concern if C wanted it with different features or semantics.

Which would make compatibility more difficult.
SG-16 proposal
Questions:
Jens: Basically, a good idea. Not sure that C will have the bandwidth for C23. Might be that the two other papers are higher priority.
Aaron: For background, working on background for a WG14 paper for this. Even if WG14 doesn't want to adopt the feature but get some sort of status quo that we wouldn't want to get a different syntax to make sure the two languages don't go in orthogonal directions.
Aaron: No incompatibilities except with a compiler Philip is expert in. He will check and respond.
Phillip: SDCC treats \o as o (AFAIK for any \X that is not defined by the standard, SDCC treats \X as X). I do not know if any users rely on that, but I don't think so (will still ask on the sdcc-user list).
Clive: Is the e coincidental with the closing brace?
Corentin: It's just part of the example. Missing closing brace would be a constraint violation
Jens: Only x and u would be constraint violations
Aaron: Was implemented in Clang yesterday so not much user experience
Aaron: Any polls? Maybe just get the feel from C folks
Jens: Low priority
Philip: Allow a classical hex prefix inside the curly braces?
Corentin: Not sure what the motivation should be
Philip: Add a sentence on why no hex in the braces
Aaron: The primary motivation is to make sure the lexers do the same thing to keep lexing and preprocessing in step between C and C++ compilers. Motivation for users
Tom: Having it different in C and C++ would be much of a deal because you still need to show older versions.
Aaron: Because this is constraint violation an implementation is free to provide it as an extension for backwards compatibility
Ville: Polling would be useful to get some sort of record.

POLL: Are the WG14 members of SG22 in favor of Wg14 adopting the same syntax and semantics as WG21 P2290R1 for some future version of the C Standard?

Committee	For	Against	Abstain	Notes
WG14	5	0	2	Consensus

P2316R0 Consistent character literal encoding

Corentin Jabot presenting
Status Quo is that constant expressions in the #if directive are different than expressions.
Make the literal encoding in the preprocessor and in phase 5+ the same
C should adopt this but if C say the encoding maybe different and C++ says they are not it doesn't create an incompatibility
External preprocessing not a real concern could add a flag to specify the execution encoding to preprocessors
Evolution approved unanimously
Implementation
Corentin: Nothing here.

Jens: this kind of thing is kind of rare. Only looked through open-source projects. Ask some questions.
Corentin: This C++ paper will have no effect on any existing compilers.
Tom: Standards have external preprocessors and don't get to control what the C Compiler sees. The motivation is if the preprocessor uses one encoding and the compiler uses a different encoding.
Aaron: Does anyone know of C compilers that have separate preprocessors?
Ville: There is a compiler that has the option of running a separate preprocessor. There is no code breakage concern. It is just sanctioning what compilers do today.
Philip. SDCC uses a separate preprocessor. Could try to find out if people do weird things for character sets.
Aaron: It would be interesting to see if this problem does cause a compatibility and taking source code that was working by chance and possibly break it.
Philip: Could break code that relies on implementation-defined behavior.
Ville: I suppose it's worth pointing out. For most compilers, nothing needs to be done to implement this.
Will: Boost has a separate preprocessor implementation.
Corentin: If someone wanted to write a C version of this paper it would be a good idea.
Aaron: Probably not time for C23 but maybe afterwards.

POLL: Are the WG14 members of SG22 in favor of something along the lines of P2316R0 in C (no time frame specified)?

Committee	For	Against	Abstain	Notes
WG14	6	0	1	Consensus

P2295R4 Support for UTF-8 as a portable source file encoding

Corentin Jabot presenting
Corentin: Do not specify a preferred source file, so it's impossible to specify a portable source file. This paper says UTF-8 source files should be supported by all C++ compilers.
Jens: Version being projected is different than the one we have seen.
Corentin: It's possible that we're looking at a new revision. We're looking at F5. Just arrived at this wording on Wednesday of this week.
Philip: What force an implementation to support an additional flag?
Corentin: If it only supports UTF-8 it doesn't need to add an additional flag.
Aaron: How much source code is this going to break.
Corentin: Implementation could consider UTF-8 encoding. If you just provide a flag. Not intended to break any code.
Aaron: Standard doesn't get to specify flags.
Tom: An implementation shall provide a means by which the encoding scheme of source files can be specified. Doesn't say what means, so a variety of means will work. One question is determining the intent of the file either by a BOM or some other source of analysis. We wanted to make support for a format distinct from how to specify it.
Corentin: We chose the warning because we said before that the means was independent of the content.
Tom: We'll talk about that later. Basically, the invocation must have a means to say what the default encoding is. An implementation can use a BOM to determine the encoding.

Ville: Similar to last paper in that UTF-8 encodings work. There is no portable way to write C and C++ code so the purpose of this is to say that there is one such encoding.
Aaron: The basic source character set was the portable encoding.
Tom: The basic source character set doesn't associate code point values with these characters.
Corentin: Doesn't say what encoding is.
Tom: Corentin has mentioned that VC source files are compiled with a UTF-8 switch which is true. In this proposal, the UTF-8 must be well formed including comments, literals, every part of the C file. It is common today to accept malformed C, many compilers don't necessarily diagnose this.
Corentin: The reason it's important
Tom: The encoding of the source code is invisible to the compiler.
Aaron: Once we understand we are a branch that we don't understand.
Jens: Weird characters need to be accepted in the preprocessor.
Aaron: #embed do characters need to be checked?
Corentin: It doesn't have to be verified because it is an array of bytes and not text.
Jens: It ends up as a long sequence of numbers.
Corentin: This is only modified phase one and after that it is business as usual.
Jens: In favor of something like this. It's overdue that we don't have a portable source code.
Corentin: Clang used UTF-8 and has been successful.

POLL: Are SG22 members in favor of something along the lines of P2295 in both C and C++?

Committee	SF	F	N	A	SA	Notes
WG14	2	4	0	0	0	Consensus
WG21	7	1	0	0	0	Consensus, author position was SF

Aaron: Probably no chance of this hitting C23?
Corentin: Probably a good chance that this makes C++23.
Corentin: Can be solved with flags.
Aaron: If C++ gets this and C doesn't could result in C++ libraries that can't import include files. There is a requirement to diagnosed malformed UTF-8.
Tom: Compilers must warn, it doesn't have to be an error, and yes there are implications to warnings.
Aaron: The introduction of new diagnostics might paint us into a corner.
Tom: The compiler running a conforming mode must produce a diagnostic.
Mark Zern: Force you to virally compile in this new mode.
Robert: Wouldn't pragmas be better since this is per include file and not.
Tom: Do this using a pragma. The IBM compiler has this. The solution probably won't use a pragma but a magic comment.
Python does magic comments, HTML does magic comments in the source file.
Corentin: Tom's paper would be a separate paper that wouldn't compete with this.

Wrapup

Aaron: Next meeting should be August 6th
Aaron: I'll schedule the next meeting and get minutes for this meeting out shortly, thanks everyone for coming.

End at 2:58 pm EST

Chat logs:

13:32:38 From philipp : SDCC treats \o as o (AFAIK for any \X that is not defined by the standard, SDCC treats \X as X). I do not know if any users rely on that, but I don't think so (will still ask on the sdcc-user list).
13:33:41 From Aaron Ballman : Thank you, Philipp!
13:55:22 From Will Wray : FYIs
Boost has a separate preprocessor implementation
https://github.com/boostorg/wave
https://www.boost.org/doc/libs/1_76_0/libs/wave/