

Document: WG14/N1295
Date: 2008/03/14
Project: WG14 TR 24732
References: WG14/N1216, WG14/N1270
Authors: Rich Peterson, Jim Thomas
Reply to: Rich Peterson <Rich.Peterson@hp.com>

Subject: #pragma STDC UNSUFFIXED_FLOAT_CONSTANT_IS_DECIMAL64

Problem:

At the October meeting, when N1216 was presented on removing TTDT from the TR, the committee agreed that any formulation of something like TTDT would either not meet the existing requirements for ieee-conforming floating-point behavior, or else it would involve so much analysis of the context in which a constant was used that it would present a major complication for the user to be able to know what type the compiler would choose, defeating the purpose of simplifying use. The outcome was a resolution to consider a possible standard pragma based solution.

Suggested changes to the TR:

Document page 12 (PDF 15) (the change to 6.4.4.2#4):
Change:

[4] An unsuffixed floating constant has type **double**.

to:

[4] An unsuffixed floating constant has type **double**, unless modified by the standard pragma UNSUFFIXED_FLOAT_CONSTANT_IS_DECIMAL64

Document pages 13-15 (PDF 16-18):

After new paragraph [4a], add another new paragraph:

[4b] The type given to an unsuffixed floating-point constant is normally type double. However, the following pragma may be used to change this behavior:

```
#pragma STDC UNSUFFIXED_FLOAT_CONSTANT_IS_DECIMAL64 [ON|OFF]
```

This pragma directs the implementation to treat unsuffixed floating-point constants as having type double (when the `OFF` keyword is used) or type `_Decimal64` (when the `ON` keyword is used). The pragma shall occur either outside external declarations or preceding all explicit declarations and statements inside a compound statement. When outside external declarations, the pragma takes effect from its occurrence until another UNSUFFIXED_FLOAT_CONSTANT_IS_DECIMAL64 pragma is encountered, or until the end of the translation unit. When inside a compound statement, the pragma takes effect from its occurrence until another UNSUFFIXED_FLOAT_CONSTANT_IS_DECIMAL64 pragma is encountered (including within a nested compound statement), or until the end of the compound statement; at the end of a compound statement the state for the pragma is restored to its condition just before the compound statement. If this pragma is used in any other context, the behavior is undefined. The default setting of this pragma is `OFF`.

Remove section 7.1.1 Translation time data type, and replace by the following:

7.1.1 pragma UNSUFFIXED_FLOAT_CONSTANT_IS_DECIMAL64

Source code that uses both generic and decimal floating point values in close proximity ought to use the suffixed forms of floating-point constants for clarity. However, it may be expected that a typical usage pattern would be that within significant portions of source code, all of the floating-point usage would be purely generic or purely decimal. The pragma introduced in section 7.1 allows programmers to establish a context in which unsuffixed floating-point constants would be uniformly interpreted as having either type double (as they do in C99) or type `_Decimal64`.

Suggested changes to C99:

In 6.10.6, paragraph 2, add to the list of pragmas:

```
#pragma STDC UNSUFFIXED_FLOAT_CONSTANT_IS_DECIMAL64 [ON|OFF]
```