WG14/N1524
October 11 2010
Thomas Plum, tplum@plumhall.com

Using specifier, not qualifier, for _Atomic

I still have the opinion that _Atomic(type-name) is the only syntax we need for atomics, and that it's over-specifying to require every implementation to support the _Atomic qualifier.

To the best of my knowledge, the underlying issues are entirely a question of aesthetics and taste; I don't mean to minimize those issues, but WG14 can consider them at the Batavia meeting.

Here are the details; I think this covers every change needed ...

Changes to 6.2.5 Types:

append to para 20:
-- An atomic type describes the type designated by the construct _Atomic(type-name). (Atomic types are a conditional feature that implementations need not support; see 6.10.8.)

delete para 27:
27 Further, there is the _Atomic qualifier, which may combine with volatile and restrict. The size, representation, and alignment of an _Atomic-qualified type need not be the same as those of the corresponding unqualified type. (Atomic types are a conditional feature that implementations need not support; see 6.10.8.)

Globally change "_Atomic-qualified type" to "atomic type".

Delete _Atomic from 6.7.3 para 1 "type qualifier:"

6.3.2.1 Lvalues, arrays, and function designators
says
"If the lvalue has qualified type, the value has the unqualified version of the type of the lvalue; otherwise, the value has the type of the lvalue." When _Atomic is no longer a qualifier, then we need to add "If the lvalue has qualified type, the value has the unqualified version of the type of the lvalue; if the lvalue has atomic type, the value has the non-atomic version of the type of the lvalue;
otherwise, the value has the type of the lvalue."

6.7.3 para 5 says:
"If an attempt is made to refer to an object defined with an _Atomic-qualified type through use of an lvalue with non-_Atomic-qualified type, the behavior is undefined."

This needs to be moved, e.g. to 6.3.2.1 and revised to say "If an attempt is made to
refer to an object defined with an atomic type through use of an lvalue with non-atomic type, the behavior is undefined."