N1863: Proposed resolution of DR423 *underspecification for qualified rvalues*

Blaine Garst  
September 20, 2014

The committee reached substantial agreement on this DR in October of last year, with only the issue of whether `_Atomic` muddied the waters. N1803 was an aborted attempt to simplify the specification of `_Atomic`, but was withdrawn in the face of conflicting field practice.

What is well established at this point is that

1) casts drop qualifiers (even if named)  
2) functions return unqualified types

The key remaining issue is whether `_Generic` selection should apply to `_Atomic` qualified types. This now seems straightforward, since the committee already has agreed that the controlling expression of generic selection cannot have qualified type.

As such, I propose the following:

**Suggested Technical Corrigendum**

Change 6.5.4.p5 from

> Preceding an expression by a parenthesized type name converts the value of the expression to the named type. This construction is called a *cast*.104) A cast that specifies no conversion has no effect on the type or value of an expression.

104) A cast does not yield an lvalue. Thus, a cast to a qualified type has the same effect as a cast to the unqualified version of that type.

To

> Preceding an expression by a parenthesized type name converts the value of the expression to the unqualified version of the named type. This construction is called a *cast*.104) A cast that specifies no conversion has no effect on the type or value of an expression.

104) A cast does not yield an lvalue.

Change 6.7.6.3 p5 from
... then the type specified for `ident` is "derived-declarator-type-list function returning T"

to

... then the type specified for `ident` is "derived-declarator-type-list function returning the unqualified version of T"