

Attribute Syntax

Nick Stoughton

At the Kona 2007 WG 14 meeting, WG 14 reviewed a number of papers outlining attributes for C. These included the proposals from C++ (WG 14 N1262 or WG 21 N2418), and a liaison statement from WG 14 to WG 21 (WG 14 N1273) was delivered and discussed at the WG 21 Bellevue 2008 meeting.

WG 14 could not agree on a syntax at the Kona meeting. However, the group agreed that:

- whatever syntax is used, it must be compatible with existing practice, to the extent that simple `#define` macros could be used to move from preexisting practice syntax (e.g. `__attribute__((x))` or `__declspec(x)`) to the new syntax and vice versa. Attributes must be permitted in the position of a storage class specifier, i.e. an attribute applies to all declarators.
- the existing C `_Pragma` keyword goes a long way toward solving the generalized issue, and might well serve as the ultimate way of expressing attributes at levels not currently recognized in existing practice (e.g. at block or translation unit level).

WG 21 accepted much of this criticism and modified their proposal to permit attributes in the position of a storage class specifier so that the attribute applies to all declarators. However, they strongly felt that the “[]” notation was important. The liaisons present at the time (Nick Stoughton, Barry Hedquist, and Clark Nelson) agreed that they would represent this compromise to WG 14.

It is the expectation that the “[]” notation will be in shipping C++ compilers shortly, and certainly by the time that the C1X revision is complete. To that end, it appears that it should be possible to meet the “existing practice” criteria of WG 14. However, no detailed syntax proposal is ready at this time; it is expected that such a paper could be submitted either at the Fall 2008 or Spring 2009 WG 14 meeting.