

Wording for `[[maybe_unused]]` attribute.

Document No.: P0212R1

Revises: P0212R0

Project: Programming Language C++ - Evolution

Author: Andrew Tomazos <andrewtomazos@gmail.com>

Date: 2016-03-01

Summary

A wording for the `[[maybe_unused]]` attribute is proposed for application to the C++17 Working Draft. The attribute is described in the previous revision.

Wording

7.6.6 Maybe unused attribute

`[dcl.attr.unused]`

1. The *attribute-token* `maybe_unused` indicates that a name or entity is possibly intentionally unused. It shall appear at most once in each *attribute-list* and no *attribute-argument-clause* shall be present.
2. The attribute may be applied to the declaration of a class, a *typedef-name*, a variable, a non-static data member, a function, an enumeration, or an enumerator.
3. [Note: For an entity marked `maybe_unused`, implementations are encouraged not to emit a warning that the entity is unused, or that the entity is used despite the presence of the attribute.]
4. A name or entity declared without the `maybe_unused` attribute can later be redeclared with the attribute and vice versa. An entity is considered marked after the first declaration that marks it.

5. [Example:

```
[[maybe_unused]] void f([[maybe_unused]] bool thing1,  
                          [[maybe_unused]] bool thing2) {  
    [[maybe_unused]] bool b = thing1 && thing2;  
    assert(b);  
}
```

Implementations are encouraged not to warn that `b` is unused, whether or not `NDEBUG` is defined.]