

Core issue 968: Disambiguating "[[" (revision 1)

Notes

In Santa Cruz we decided that the sequence "[[" should follow a max-munch principle and always be interpreted as the beginning of attributes. I don't think we discussed whether that meant:

- (a) having "[[" always start an *attribute-specifier* in any context, or
- (b) having "[[" in contexts that allow an *attribute-specifier*.

I decided to go with the former option, because we may want to allow attributes in more contexts in the future (e.g., in *new-type-ids*, as in the example of the proposed wording), and the only alternative—subscript expressions starting with a lambda-expression—are unlikely and can be parenthesized to avoid the conundrum.

The changes are against N3035.

Wording Changes

In 7.6.1 [dcl.attr.grammar] append the following paragraph:

- 6 Two consecutive left square bracket tokens shall appear only when introducing an *attribute-specifier*. [*Note*: If two consecutive left square brackets appear where an *attribute-specifier* is not allowed, the program is ill-formed even if the brackets match an alternative grammar production. —*end note*] [*Example*:

```
int p[10];
void f() {
    int x = 42, y[5];
    int(p[[x]{return x;}()]); // Error: malformed attribute on a nested
                             // declarator-id and not a function-style cast of
                             // an element of p.
    y[[]]{return 2;}() = 2; // Error even though attributes are not allowed
} // in this context.
—end example ]
```