Core issue 951: Various Attribute Issues  
(revision 1)

Notes

The write-up of 951 suggests allowing attributes in "a for-init-statement that is an expression-statement" or preceding various compound-statements: I did not follow that suggestion since there is no existing parallel for it. I also did not follow the suggestion to allow attributes preceding a type-specifier-seq in a type-id, since in other contexts prefix attributes appertain to the declarator-id entity, and there is no declarator-id in this case.

A trailing optional attribute-specifier has been folded into decl-specifier-seq and type-specifier-seq to simplify the overall wording (and avoiding oversights in the future).

In a few places the location of the optional attribute-specifier has been moved to make it consistent with similar uses elsewhere.

The changes are against N3035. In some places the changes overlap with changes for core issues 743/950 and 962 (the latter is in ready status).

Wording Changes

In 5.3.4 [expr.new] paragraph 1 amend the grammar rule for noptr-new-declarator as follows (to match its noptr-abstract-declarator counterpart):

\[
noptr-new-declarator:  
\begin{align*}  
[ \text{expression} ] & \text{attribute-specifier}_\text{opt} \\
& \text{noptr-new-declarator} [ \text{constant-expression} ] \text{attribute-specifier}_\text{opt}  
\end{align*}
\]

In 5.3.4 [expr.new] paragraph 5, append the following sentence:

... The attribute-specifier in a noptr-new-declarator appertains to the associated array type.

In 6.4 [stmt.select] paragraph 1 amend the grammar rule for condition as follows:

\[
\begin{align*}  
\text{condition}:  
\text{expression}  \\
\text{attribute-specifier}_\text{opt} \text{type-specifier-seq} \text{attribute-specifier}_\text{opt} \text{declarator} = \text{initializer-clause}  \\
\text{attribute-specifier}_\text{opt} \text{type-specifier-seq} \text{attribute-specifier}_\text{opt} \text{declarator} \text{braced-init-list}  
\end{align*}
\]

In 6.5 [stmt.iter] paragraph 1 amend the grammar and text as follows:

...
for-range-declaration:
  expression
  attribute-specifier_opt type-specifier-seq attribute-specifier_opt declarator

See 8.3 [dcl.meaning] for the optional attribute-specifier in a for-range-declaration.
[Note:...]

In 7 [dcl] paragraph 1 amend the grammar rule for simple-declaration as follows:
simple-declaration:
  attribute-specifier_opt decl-specifier-seq_opt attribute-specifier_opt
  init-declarator-list_opt ;

and amend the text that follows as indicated:

... The simple-declaration
  attribute-specifier_opt decl-specifier-seq_opt attribute-specifier_opt
  init-declarator-list_opt ;

is divided into four three parts. Attributes are described in 7.6 [dcl.attr]. decl-specifiers, the principal components of a decl-specifier-seq, are described in 7.1. The two optional attribute-specifiers and declarators, the components of an init-declarator-list, are described in Clause 8. The optional attribute-specifier in a simple-declaration appertains to each of the entities declared by the declarators; it shall not appear if the optional init-declarator-list is omitted. [ Note: In the declaration for an entity, attributes appertaining to that entity may appear both at the start of the declaration and after the declarator-id for that declaration. —end note ][ Example: 
  [[noreturn, nothrow]] void f [[noreturn]] () ; // OK
—end example ]

In 7 [dcl] paragraph 9, delete the second sentence:

... If it is omitted, an attribute-specifier shall not appear.

In 7.1 [dcl.spec] paragraph 1 amend the grammar and text as follows:

... decl-specifier-seq:

  decl-specifier-seq_opt decl-specifier
  decl-specifier attribute-specifier_opt
  decl-specifier decl-specifier-seq

The optional attribute-specifier in a decl-specifier-seq appertains to the type determined by the decl-specifier-seq (8.3 [dcl.meaning]). The attribute-specifier affects the type only for the declaration it appears in, not other declarations involving the same type.

In 7.1.6 [dcl.type] paragraph 1 amend the grammar and text as follows:
... type-specifier-seq:
  type-specifier type-specifier-seq
  type-specifier attribute-specifier
  type-specifier type-specifier-seq

trailing-type-specifier-seq:
  trailing-type-specifier trailing-type-specifier-seq
  trailing-type-specifier attribute-specifier
  trailing-type-specifier trailing-type-specifier-seq

The optional attribute-specifier in a type-specifier-seq or trailing-type-specifier-seq appertains to the type denoted by the preceding type-specifiers (8.3 [dcl.meaning]). The attribute-specifier affects the type only for the construct it appears in, not other constructs involving the same type.

In 7.2 [dcl.enum] paragraph 1 amend the grammar and text as follows:
...
enum-head:
  enum-key attribute-specifier
  enum-base attribute-specifier
  enum-key attribute-specifier
  identifier
  enum-key attribute-specifier
  nested-name-specifier identifier
  enum-base attribute-specifier

opaque-enum-declaration:
  enum-key attribute-specifier
  identifier
  enum-base attribute-specifier
  ...

The first optional attribute-specifier in the enum-head and the opaque-enum-declaration appertains to the enumeration; the attributes in that attribute-specifier are thereafter considered attributes of the enumeration whenever it is named. The second optional attribute-specifier in the enum-head and the opaque enum-declaration shall appear only if the enum-base is present; it appertains to the enum-base.

In 8 [dcl.decl] paragraph 2 amend the first sentence as follows (note also the added comma):
2 The two three components of a simple-declaration are the attributes (7.6 [dcl.attr]), the specifiers (decl-specifier-seq; 7.1), and the declarators (init-declarator-list). ...

In 8 [dcl.decl] paragraph 4 amend the grammar rule for trailing-return-type and ptr-operator as follows:
trailing-return-type:
  -> attribute-specifier
  -> trailing-type-specifier-seq
  -> abstract-declarator
ptr-operator:
In 8.1 [dcl.name] paragraph 1 amend the grammar rule for `type-id` as follows:

```
type-id:
  type-specifier-seq attribute-specifier_opt abstract-declarator_opt
```

In 8.3 [dcl.meaning] paragraph 3 amend the following phrase as indicated:

```
... of the form attribute-specifier_opt decl-specifier-seq attribute-specifier_opt and ...
```

In 8.3 [dcl.meaning] amend paragraph 5 as follows:

```
5 In a declaration `attribute-specifier_opt T attribute-specifier_opt D` where `D` is an unadorned identifier the type of this identifier is “T”. The first optional `attribute-specifier` appertains to the entity being declared. The second optional `attribute-specifier` appertains to the type `T`, but not to the class or enumeration declared in the `decl-specifier-seq`, if any.
```

In 8.3.2 [dcl.ref] amend paragraph 1 as follows:

```
1 In a declaration `T D` where `D` has either of the forms
   & attribute-specifier_opt D1
   && attribute-specifier_opt D1

and the type of the identifier in the declaration `T D` is “derived-declarator-type-list T,” then the type of the identifier of `D` is “derived-declarator-type-list reference to T.” The optional `attribute-specifier` appertains to the reference type.
```

In 8.3.5 [dcl.fct] paragraph 1 amend the grammatical form as follows:

```
D1 ( parameter-declaration-clause ) attribute-specifier_opt cv-qualifier-seq_opt
   ref-qualifier_opt exception-specification_opt attribute-specifier_opt
```

In 8.3.5 [dcl.fct] paragraph 2 amend the grammatical form as follows:

```
D1 ( parameter-declaration-clause ) attribute-specifier_opt cv-qualifier-seq_opt
   ref-qualifier_opt exception-specification_opt attribute-specifier_opt
   trailing-return-type
```

In 8.3.5 [dcl.fct] paragraph 3 amend the grammar follows:
...  

\[
\text{parameter-declaration:} \quad \begin{array}{c}
\text{attribute-specifier}_{\text{opt}} \text{ decl-specifier-seq attribute-specifier}_{\text{opt}} \text{ declarator} \\
\text{attribute-specifier}_{\text{opt}} \text{ decl-specifier-seq attribute-specifier}_{\text{opt}} = \text{assignment-expr} \\
\text{attribute-specifier}_{\text{opt}} \text{ decl-specifier-seq attribute-specifier}_{\text{opt}} \quad \text{abstract-declarator}_{\text{opt}} \\
\text{attribute-specifier}_{\text{opt}} \text{ decl-specifier-seq attribute-specifier}_{\text{opt}} \quad \text{abstract-declarator}_{\text{opt}} = \text{assignment-expr} 
\end{array}
\]

and append the following text:

The optional attribute-specifier in a parameter-declaration appertains to the parameter.

In 8.4 [dcl.fct.def] paragraph 1, amend the grammar rule for function-definition as follows:

\[
\text{function-definition:} \quad \begin{array}{c}
\text{attribute-specifier}_{\text{opt}} \text{ decl-specifier-seq attribute-specifier}_{\text{opt}} \quad \text{declarator} \quad \text{function-body} \\
\text{attribute-specifier}_{\text{opt}} \text{ decl-specifier-seq attribute-specifier}_{\text{opt}} \quad \text{declarator} = \text{default} \\
\text{attribute-specifier}_{\text{opt}} \text{ decl-specifier-seq attribute-specifier}_{\text{opt}} \quad \text{declarator} = \text{delete} 
\end{array}
\]

and append the following sentence at the end of the paragraph:

The optional attribute-specifier in a function-definition appertains to the function.

In 8.4 [dcl.fct.def] paragraph 2, amend the grammatical form as follows:

\[
\text{D1} (\text{parameter-declaration-clause}) \text{ cv-qualifier-seq}_{\text{opt}} \text{ ref-qualifier}_{\text{opt}} \\
\text{exception-specification}_{\text{opt}} \text{ attribute-specifier}_{\text{opt}} \text{ trailing-return-type}_{\text{opt}}
\]

In 8.4 [dcl.fct.def] paragraph 9, amend the grammatical form as follows:

\[
\text{attribute-specifier}_{\text{opt}} \text{ decl-specifier-seq attribute-specifier}_{\text{opt}} \text{ declarator} = \text{default} 
\]

In 8.4 [dcl.fct.def] paragraph 10, amend the grammatical form as follows:

\[
\text{attribute-specifier}_{\text{opt}} \text{ decl-specifier-seq attribute-specifier}_{\text{opt}} \text{ declarator} = \text{delete} 
\]

In the introduction of 9.2 [class.mem], amend the first production in the grammar rule for member-declaration as follows:
member-declaration:

attribute-specifier<opt> decl-specifier-seq<opt> attribute-specifier<opt>
member-declarator-list<opt> ;

...

Insert the following new paragraph after 9.2 [class.mem] paragraph 6:

6b The optional attribute-specifier in a member-declaration appertains to each of the entities declared by the member-declarators; it shall not appear if the optional member-declarator-list is omitted.

In 10 [class.derived] paragraph 1 amend the grammar rule for base-specifier as follows:

debase-specifier:

attribute-specifier<opt> ::<opt> attribute-specifier<opt> virtual<opt> access-specifier<opt>

::<opt> nested-name-specifier<opt> class-name attribute-specifier<opt>

attribute-specifier<opt> access-specifier virtual<opt>

::<opt> nested-name-specifier<opt> class-name attribute-specifier<opt>

In 12.3.2 [class.conv.fct] paragraph 1 amend the grammar rule for conversion-type-id as follows:

debconversion-type-id:

type-specifier-seq attribute-specifier<opt> conversion-declarator<opt>

In 15 [except] paragraph 1 amend the grammar and text as follows:

... exception-declaration:

attribute-specifier<opt> type-specifier-seq declarator
attribute-specifier<opt> type-specifier-seq abstract-declarator<opt>

... type-specifier-seq

...

The optional attribute-specifier in an exception-declaration appertains to the formal parameter of the catch clause (15.3 [except.handle]).