Slay Some Earthly Demons IV
Document: n3342
Author: Martin Uecker
Date: 2024-09-08

This remaining issues from n3244 were split up into the following series:

n3340 - Slay Some Earthly Demons II
n3341 - Slay Some Earthly Demons III
n3342 - Slay Some Earthly Demons IV
n3343 - Slay Some Earthly Demons V
n3344 - Slay Some Earthly Demons VI
n3345 - Slay Some Earthly Demons VII
n3346 - Slay Some Earthly Demons VIII
n3347 - Slay Some Earthly Demons IX

Meeting meeting notes about previous discussion can be found in n3281. The following numbers correspond to N3220 and the ones with ($\rightarrow$) to N3301.

J.2.19 needs more work, cf. SC22WG14.25600 (**note to editor:** parts applied by mistake)
J.2.21 done (incl. additional change), (**note to editor:** there is a "not" missing in 6.3.2.1)
n3340 J.2.40 ($\rightarrow$ J.2.38) withdrawn in June, now revised (cf. SC22WG14.25600)
J.2.56 done
J.2.57 done (option 1)
n3341 J.2.58 ($\rightarrow$ J.2.54) not discussed in June
J.2.60 ($\rightarrow$ J.2.56) direction to editor (**note to editor:** not yet done)
n3342 J.2.63 ($\rightarrow$ J.2.59) no consensus in June, now revised to make implementation-defined
n3343   wording change not related to an entry in Annex J
J.2.67 done
J.2.69 done (alternative wording for semantic section)
n3344 J.2.75 ($\rightarrow$ J.2.70) now revised (cf. SC22WG14.25600)
J.2.77 ($\rightarrow$ J.3.13. 2) not discussed in June, but fixed editorially
J.2.78 ($\rightarrow$ J.3.13. 3) not discussed in June, but fixed editorially
n3345 J.2.79 ($\rightarrow$ J.2.72) not discussed in June
n3346 J.2.80-J.2.82 ($\rightarrow$ J.2.73-J.2.75) not discussed in June (cf. SC22WG14.25600)
n3347 J.2.87 ($\rightarrow$ J.2.80) not discussed in June

<u>Annex J</u> (J.2.59 N3301, J.2.63 N3220)

(59) The specification of a function type includes any type qualifiers (6.7.4).

<u>Example</u>

https://godbolt.org/z/vffs4dG9c

```
typedef int f();

const f g;
```

<u>Analysis</u>

Clang/MSVC ignore such qualifiers with a warning. GCC silently except in pedantic mode. This should be a constraint violation to detect mistakes and / or so it could potentially be used in the future (e.g. wording for N3089 could modify this again). In the June 2024 meeting there was no consensus for making this a constraint violation. It is now suggested to make this implementation-defined.

<u>Proposed Change</u> (**update:** change to implementation-defined)

Semantics

10 If the specification of an array type includes any type qualifiers, both the array and the element type are so-qualified. If the specification of a function type includes any type qualifiers, the behavior is ~~undefined~~ **implementation-defined.**151)

Unrelated editorial grammar fix:

51) This can occur with typedef s. This rule does not apply to the _Atomic qualifier.~~, and that Q~~ **Q**ualifiers do not have any direct effect on the array type itself, but affect conversion rules for pointer types that reference an array type.