Proposal for C23
WG14 N 3112

Title:	Compatible Types
Author, affiliation:	Robert C. Seacord, Woven Planet
Date:	2023-2-13
Proposal category:	Defect
Target audience:	Implementers
Abstract:	Compatible types
Prior art:	C

Compatible Types

Reply-to: Robert C. Seacord (rcseacord@gmail.com)
Document No: **N 3112**
Reference Document: **N 3019**
Date: 2023-2-17

Change Log

2023-2-17:

- Initial version

1.0 Introduction and Rationale

The aliasing rules Subclause 6.5, paragraph 7 allows “the signed or unsigned type” to alias another object but fails to mention type compatibility. Consequently, it is unclear if a compatible enumeration type can alias such an object.

2.0 Proposed Solution

The proposed solution is to allow compatible types to alias in all cases.

3.0 Wording

Subclause 6.5, “Expressions”, paragraph 7:

Change:

— a type that is the signed or unsigned type corresponding to the effective type of the object,

— a type that is the signed or unsigned type corresponding to a qualified version of the effective type of the object,

to:

— the signed or unsigned type compatible with the underlying type of the effective type of the object,

— the signed or unsigned type compatible with a qualified version of the underlying type of the effective type of the object,

Make the following changes to subclause 7.16.1.1, “The va_arg macro”, paragraph 2:

– one type is compatible with a signed integer type, the other type is compatible with the corresponding unsigned integer type, and the value is representable in both types;

4.0 Acknowledgements

I would like to recognize the following people for their help with this work: Martin Uecker, Jens Gustedt, and Joseph Myers.

5.0 References

None.