

Document Number: WG14 N505/X3J11 95-106

## C9X Revision Proposal

=====

Title: Make qualifiers idempotent  
 Author: Clive D.W. Feather  
 Author Affiliation: CityScape Internet Services Ltd.  
 Postal Address: 9 Covent Garden, Cambridge, UK, CB1 2HR  
 E-mail Address: cdwf@cityscape.co.uk or clive@demon.net  
 Telephone Number: +44 1223 566950  
 Fax Number: +44 1223 566951  
 Sponsor: BSI  
 Date: 1995-12-05

## Proposal Category:

- ☐ Editorial change/non-normative contribution
- ☐ Correction
- ☐ New feature
- ☐ Addition to obsolescent feature list
- ☐ Addition to Future Directions
- ☒ Other (please specify) removes unnecessary restriction

## Area of Standard Affected:

- ☐ Environment
- ☒ Language
- ☐ Preprocessor
- ☐ Library
- ☐ Macro/typedef/tag name
- ☐ Function
- ☐ Header
- ☐ Other (please specify) \_\_\_\_\_

Prior Art: n/a

Target Audience: n/a

Related Documents (if any): none

Proposal Attached: Yes

## Abstract:

This proposal allows a type to have more than one occurrence of the same qualifier; the effect of this is the same as having it only once.

## Notes:

This proposal was prepared with the assistance of Mark Brader, Jutta Degener, Ron Guilmette, and a person whose employment conditions require anonymity.

Proposal UK008a - Make qualifiers idempotent

=====

## Summary

-----

This proposal allows a type to have more than one occurrence of the same qualifier; the effect of this is the same as having it only once.

## Conformance

-----

No strictly-conforming application is affected by this proposal. Some applications which formerly generated a diagnostic become strictly conforming

## Discussion

---

C89 allows a type to be qualified only once by a given qualifier. If the qualifier is hidden in a typedef in a header, it can be almost impossible to determine whether the type is qualified other than by trial and error. This proposal allows multiple qualification with no effect. For example, `sig_atomic_t` could now be made a volatile type.

## Detailed proposal

---

In subclause 6.5.3, delete the constraints, and add a new paragraph to the semantics:

If the same qualifier appears more than once in the same specifier list or qualifier list, either directly or via one or more typedefs, the effect shall be the same as if it appeared only once.