

Clarifications to Anonymous Structures and Unions

David Keaton

2010-12-22

1. Introduction

1.1 Purpose

This proposal specifies wording changes to the language portion of the C standard to clarify the intended meaning of the section describing anonymous structures and unions.

1.2 Scope

This document falls within the scope of the C standard, and thus follows all rules and guidelines of that standard except where explicitly noted herein. All proposed changes are relative to WG14/N1539.

1.3 References

1. ISO/IEC 9899:1999(E), *Programming Languages—C*.
2. WG14/N1539, Committee Draft of C1X, 2010-11-16.
3. Anonymous structs questions, Joseph Myers, <<http://www.open-std.org/jtc1/sc22/wg14/12205>>.

1.4 Rationale

Joseph Myers pointed out that the current wording of the C standard could be interpreted to mean that a typedef-name could be used to declare an anonymous structure or union. This was not intended because it is a gratuitous difference from C++.

Joseph also pointed out that some of the wording makes it unclear whether a flexible array member can be preceded by only anonymous structures and unions. The confusion occurs because a flexible array member must be preceded by a “named member.” It was intended that anonymous structures and unions bring named members into the scope of their parent structure or union (recursively).

The following wording is proposed to eliminate these problems.

2. Language

Changed text surrounded by unchanged text is underlined in the following section.

2.1 Changes to subclause 6.7.2.1

In paragraph 8, change the third sentence to the following.

If the struct-declaration-list does not contain any named members, either directly or via anonymous structures or anonymous unions, the behavior is undefined.

In paragraph 13, change the first sentence to the following.

An unnamed member whose type specifier is a structure specifier with no tag is called an *anonymous structure*; an unnamed member whose type specifier is a union specifier with no tag is called an *anonymous union*.