WG14, N3650

WG14's C Indentation and Brace Styles

David Svoboda
svoboda@sei.cmu.edu
Date: 2025-07-14

The Question

C code can be programmed using a variety of styles for indentation and braces. Wikipedia has a good overview: https://en.wikipedia.org/wiki/Indentation_style (accessed 2025-7-14)

Sometimes C23 uses the "One True Brace Style" (1TBS), where a function's opening brace as the same line as its declaration. For example, here is the code in (n3220) s6.3.1.8p3:

```
_BitInt(2) a2 = 1;
_BitInt(3) a3 = 2;
_BitInt(33) a33 = 1;
signed char c = 3;

a2 * a3; /* As part of the multiplication, a2 is converted to
            _BitInt(3) and the result type is _BitInt(3). */
a2 * c;  /* As part of the multiplication, c is promoted to int,
            a2 is converted to int and the result type is int. */
a33 * c; /* As part of the multiplication, c is promoted to int.
            Then, provided int has a width of at most 32,
            it is converted to _BitInt(33) and the result type
            is _BitInt(33). */

void func(_BitInt(8) a8, _BitInt(24) a24) {
      /* Cast one of the operands to 32-bits to guarantee the
         result of the multiplication can contain all possible values. */
      _BitInt(32) a32 = a8 * (_BitInt(32))a24;
}
```

At other times, C23 uses the "Kernighan & Ritchie" (K&R) style, where a function's opening brace is the sole character in the line following its declaration. For example, here is the code in (n3220) s6.4.2.2p3:

```
#include <stdio.h>
void myfunc(void)
{
      printf("%s\n", __func__);
      /* ... */
}
```

People can and often do disagree on which coding style should be used, and this is considered acceptable; conventional wisdom does not prefer any single coding style; it only recommends that a codebase should have one style enforced consistently throughout the code. Guidelines that focus on

“deeper" issues like maintainability or security avoid discussing style. Most coding styles can be enforced automatically by static analysis tools or code pretty-printing software.

As a single data point, the SEI CERT C Coding Standard usually complies with the 1TBS. But it makes exceptions when citing source code from other sources, such as the Linux kernel or the Windows DCOM API. In these exceptions we preserve the original spacing.

WG14 has historically left decisions such as formatting of the code to its document editor and the committee seems to consider coding style questions as out-of-scope. An informal inquiry suggests there is no official convention or documentation about coding style, or how code is to be formatted within the standard itself.

Does WG14 wish to use an internal coding style consistently (whether we explicitly promote this style or not)?

Proposals

As coding style has been historically left to the editor's discretion, any decision WG14 makes would take the form of instructions to the editor, rather than normative text to be applied to the standard. The committee could choose one of the following:

1. Enforce the 1TBS style in all code examples throughout the document.
2. Enforce the K&R style in all code examples throughout the document.
3. Enforce no style; leave it completely at the editor's discretion.
4. Enforce no style but instruct the editor to use a variety of coding styles throughout the document. (This would indicate WG14's neutrality towards coding styles).

Questions

Should the ISO C Introduction introduce any normative text regarding the stylistic layout of code? Should it address brace styles or indentation?

Acknowledgements

We originally discovered the ISO C current examples in https://github.com/sei-dsvoboda/exub/pull/2#issuecomment-3016840414

Thanks to Chris Bazley and the Undefined Behavior Study Group