# Relocate Endian's Specification

## Contents

### Abstract

This paper proposes to relocate the library's **enum class endian** facility from header **<type_traits>** to the recently-created **<bit>** header.

> *How do you open a soft boiled egg? Big-Endian or Little-Endian? Whack or tap?*
>
> — DANA VELDEN

> *It is allowed on all hands, that the primitive way of breaking eggs, before we eat them, was upon the larger end; but his present majesty's grandfather, while he was a boy, going to eat an egg, and breaking it according to the ancient practice, happened to cut one of his fingers. Whereupon the emperor his father published an edict, commanding all his subjects, upon great penalties, to break the smaller end of their eggs. ... It is computed that eleven thousand persons have at several times suffered death, rather than submit to break their eggs at the smaller end.*
>
> — JONATHAN SWIFT

## 1  Background/proposal/discussion

The **enum class endian** was added to the C++20 Working Draft via [P0463R1]. At that time, the **<bit>** header did not yet exist, so **endian** was placed into **<type_traits>**. Since we do now have **<bit>**, it makes sense to place **endian** there (and, of course, to remove it from header **<type_traits>**). Since it has not yet shipped in any standard, there is still time to relocate **endian** to such a more suitable header.

Why relocate **endian** into **<bit>**? Because it seems clear that byteswapping and endianness are closely related. Indeed, the **byteswap** function is en route to be added, for C++20, to **<bit>** via [P1272R1] (or a future revision thereof). Also, placing **byteswap** into **<bit>** solidly indicates that the **<bit>** header has been determined to be the natural place for byte-related, as well as bit-related, functionally.

As opined during the LWG discussion of **byteswap**: Programmers seeking endianness-related facilities will naturally look in **<bit>**.

Incidentally, **enum class byte** itself has been located in header **<cstddef>** [cstddef.syn] since C++17. We do not propose to change this.

## 2 Proposed wording[1]

**2.1** Relocate the entirety of subclause [meta.endian] to the end of [bit], using [bit.endian] as the new stable name.

**2.2** Also relocate the declaration of `enum class endian` from the synopsis in [meta.type.synop] to the end of the synopsis in [bit.syn].

**2.3** Finally, insert an entry, into [tab:support.ft], with the feature-test macro name `__cpp_lib_ endian`, suitable date, and header name `<bit>`. (Note that no corresponding entry seems to have been made at the time `endian` was integrated into the Working Draft.)

## 3 Acknowledgments

Many thanks to the readers of early drafts of this paper for their thoughtful comments.

## 4 Bibliography

[N4800] Richard Smith: "Working Draft, Standard for Programming Language C++." ISO/IEC JTC1/ SC22/WG21 document N4800 (pre-Kona mailing), 2019–01–21. https://wg21.link/n4800.

[N4820] Richard Smith: "Working Draft, Standard for Programming Language C++." ISO/IEC JTC1/ SC22/WG21 document N4820 (pre-Cologne mailing), 2019–06–17. https://wg21.link/n4820.

[P0463R1] Howard Hinnant: "`endian`, Just `endian`." ISO/IEC JTC1/SC22/WG21 document P0463R1 (post-Toronto mailing), 2017–07–13. https://wg21.link/p0463r1.

[P1272R1] Isabella Muerte: "Byteswapping for fun&&nuf." ISO/IEC JTC1/SC22/WG21 document P1272R1 (post-Kona mailing), 2019–02–19. https://wg21.link/p1272r1.

## 5 Document history

| Rev. | Date | Changes |
| --- | --- | --- |
| 0 | 2019–03–10 | • Published as P1612R0, post-Kona mailing. |
| 1 | 2019–07–18 | • Rebased on [N4820]. • Use new stable tag to indicate intended table. • Fixed wording typo. • Published as P1612R1, post-Cologne mailing. |

---

[1]Proposed wording changes are based on [N4820]. Editorial instructions and drafting notes look like this .