

Document number: **P0754R0**
Date: 2017-07-30
Audience: Library Evolution Working Group
Reply to: **Alan Talbot**
cpp@alantalbot.com

<version>

In Toronto (July 2017), the removal of the vacuous <ciso646> header was discussed during the review of P0619R1 (Reviewing Deprecated Facilities of C++17 for C++20). Howard Hinnant pointed out that although the header is specified to have no effect, it is nonetheless used to determine the library version from the boilerplate front and back matter. This particular header is presumably chosen for this purpose because it is very short and therefore fast to load.

A better solution is to standardize a C++ header for this exact purpose. This header will contain only the implementation-defined boilerplate comments which specify various properties of the library such as version and copyright notice. This header will also provide a place to put other implementation-defined library meta-information which an environment or human reader might find useful. This header will contain no active code and therefore have no effect by definition.

Proposed Wording

20.5.1.2 Headers

[headers]

¶2 – Add to: Table 16 — C++ library headers

<version>

21.1 General

[support.general]

¶2 – Add to: Table 32 — Language support library summary,
21.3 Implementation properties

<version>

21.3.1 General

[support.limits.general]

New paragraph:

2 The header <version> provides a place for a conforming implementation to supply implementation-dependent information about the C++ standard library (e.g., version number and release date). Including this header has no effect.

C.2.8 Clause 20: library introduction

[diff.cpp03.library]

20.5.1.2

Change: New headers.

Rationale: New functionality.

Effect on original feature: The following C++ headers are new: <array>, <atomic>, <chrono>, <codecvt>, <condition_variable>, <forward_list>, <future>, <initializer_list>, <mutex>, <random>, <ratio>, <regex>, <scoped_allocator>, <system_error>, <thread>, <tuple>, <typeindex>, <type_traits>, <unordered_map>, ~~and~~ <unordered_set>, and <version>. In addition the following C compatibility headers are new: <ccomplex>, <cfenv>, <cinttypes>, <cstdalign>, <cstdbool>, <cstdint>, <ctgmath>, and <cuchar>. Valid C++ 2003 code that #includes headers with these names may be invalid in this International Standard.