

Business Plan and Convener's Report: ISO/IEC JTC1/SC22/WG21 (C++)

Document: ISO/IEC JTC1/SC22/WG21 N4960

Date: 2023-08-09

Period Covered: August 2022 - August 2023

Submitted By: Convener, ISO/IEC JTC1/SC22/WG21
Herb Sutter
Microsoft Corporation
1 Microsoft Way
Redmond WA 98052
Email: hsutter@microsoft.com
Tel: +1-425-707-6533

Note: The end of this annual report contains an action item jointly proposed by C, C++, and Fortran project editors.

1. MANAGEMENT SUMMARY

1.1. JTC1/SC22/WG21 STATEMENT OF SCOPE

Development and maintenance of ISO/IEC Standards, Technical Specifications, and Technical Reports related to the programming language C++.

1.2. PROJECT REPORT

1.2.1. COMPLETED PROJECTS

International Standards

- JTC1.22.14882:1998 - Programming Language C++
- JTC1.22.14882:2003 - Programming Language C++
- JTC1.22.14882:2011 - Programming Language C++
- JTC1.22.14882:2014 - Programming Language C++
- JTC1.22.14882:2017 - Programming Language C++

- JTC1.22.14882:2020 - Programming Language C++
- JTC1.22.29124:2010 - International Standard on Special Math Functions

Technical Specifications

- JTC1.22.18822:2014 - C++ Extensions for File System
- JTC1.22.19216:2018 - C++ Extensions for Networking
- JTC1.22.19217:2015 - C++ Extensions for Concepts
- JTC1.22.19568:2015 - C++ Extensions for Library Fundamentals
- JTC1.22.19568:2017 - C++ Extensions for Library Fundamentals, 2nd edition
- JTC1.22.19570:2015 - C++ Extensions for Parallelism
- JTC1.22.19570:2018 - C++ Extensions for Parallelism, 2nd edition
- JTC1.22.19571:2015 - C++ Extensions for Concurrency
- JTC1.22.19768:2007 - C++ Library Extensions
- JTC1.22.19841:2015 - C++ Extensions for Transactional Memory
- JTC1.22.21425:2017 - C++ Extensions for Ranges
- JTC1.22.21544:2018 - C++ Extensions for Modules
- JTC1.22.22277:2017 - C++ Extensions for Coroutines
- JTC1.22.23619:2021 - C++ Extensions for Reflection
- JTC1.22.24733:2011 - Extensions to Support Decimal Floating Point Arithmetic

Technical Reports

- JTC1.22.18015:2006 - C++ Performance

1.2.2. PROJECTS UNDERWAY

See isocpp.org/std/status for a summary of projects underway, including contemplated upcoming ballots.

JTC1.22.9922 - C++ Extensions for Concurrency, 2nd edition

JTC1.22.12907 - C++ Extensions for Transactional Memory, 2nd edition

JTC1.22.14882 - Work is now underway on the next revision, which is targeted for publication in 2023.

JTC1.22.19568 - C++ Extensions for Library Fundamentals, 3rd edition - Work is now underway on the next revision, which is targeted for publication in 2024.

1.2.3. CANCELLED PROJECTS

None this period.

1.2.4. COOPERATION and COMPETITION

Where appropriate, WG21 has established liaisons with other SC22 and SC22 liaison organizations' working groups. There is no direct competition with any other current SC22 working group. Occasional overlap with SC22/WG14 (C) is coordinated with regular WG21 liaison.

2. PERIOD REVIEW

2.1. MARKET REQUIREMENTS

ISO C++ remains a widely-used foundation technology, well-received in the marketplace.

Although C++ has long been a consistently popular language, since 2011 in particular it has enjoyed a renewed cycle of growth and investment in tools and platform support across the industry. This was driven primarily by the C++11 standard's completion at the same time as the industry saw a resurgence of interest in performance-efficient, hardware-efficient, and especially power-efficient systems programming capability for mobile devices, cloud data centers, high-performance financial systems, vector and GPGPU computing (via nonstandard extensions to C++ that we are now investigating standardizing), and other major growth sectors and environments.

This new cycle of industry investment in C++ includes, but is not limited to, investment in:

1. tools, such as the advent of a new major C++ implementation in the Clang compiler and other major new products actively competing to fully implement the latest ISO C++ standard;
2. organization, with the establishment of the Standard C++ Foundation trade association in 2012 (see isocpp.org/about);
3. standardization participation, so that meeting attendance is has been growing rapidly (252 experts in February 2020) organized into over 20 active subgroups -- this includes 16 active domain-specific subgroups (e.g., transactional memory, graphics, gaming) that were established since 2012 and have drawn domain experts who did not previously participate in C++ standardization; and
4. faster and more predictable standardization output, with regular releases of the standard every three years along with many concurrent Technical Specifications (14 completed and published from 2014 to 2020).

2.2. ACHIEVEMENTS

Achievements in the past year include the following.

JTC1.22.14882 - Programming Language C++ - This project was approved for publication.

2.3. RESOURCES

WG21 has grown considerably over the past three years, which reflects the continued growth and investment in C++ across the industry as noted in 2.1.

In normal (non-pandemic) times: WG21 meets three times per year in co-located technical sessions with the US committee PL22.16. WG21 regularly has experts from 20 national bodies present at meetings. WG21 has been monitoring the cross-language standards activities, and made use of the ISO/IEC JTC1/SC22 guidelines on extended characters.

WG21 has resumed having in-person meetings (with Zoom participation support) three times a year, and subgroup meetings online via Zoom dozens of times a month.

3. FOCUS NEXT WORK PERIOD

3.1. DELIVERABLES

WG21 is working on the next revision of JTC.22.14882 (IS C++) and progressing other projects as noted in 1.2.2.

3.2. STRATEGIES

WG21 members have been meeting in parallel subgroups and coordinating work between meetings via e-mail lists, teleconferences, and wiki. WG21 is working on revisions to the central IS JTC1.22.14882 on a regular three-year cadence. In addition, WG21 is parallelizing its work products by producing many work items first as Technical Specifications, which enables each independent work item to progress at its own speed and with less friction, and enables more experimental work to progress outside the main standard until it is more mature while still providing a reference for commercial implementations. When ready, these TS's can then be considered adopted (in whole or in part, and with changes) into the ISO C++ standard.

As of this writing, WG21 has over 20 active domain-specific subgroups, focusing on incubating proposals in specific areas, and which meet between WG21 face-to-face meetings via telecon and/or their own face-to-face meetings. These domain-specific groups have directly led to increased participation by leading experts in those domains who had not previously participated in WG21. For a current list of subgroups, see isocpp.org/std/the-committee.

3.3. RISKS

The COVID-19 pandemic disrupted all WG21 meetings, but we have resumed with combined in-person/online-Zoom meetings since November 2022.

3.4. OPPORTUNITIES

Nothing new to report.

3.5. WORK PROGRAM PRIORITIES

WG21 intends to continue working on new language and/or library extensions with a view to publishing multiple TSEs and then another new JTC1.22.14882 IS in 2020.

4. OTHER ITEMS

4.1. POSSIBLE ACTION REQUESTS AT FORTHCOMING PLENARY

None.

4.2 PROJECT EDITORS

The following individuals have been appointed project editors and backups.

Currently active projects:

- JTC1.22.14882, Programming Language C++:
 - Thomas Köppe (editor)
 - Jens Maurer (backup)
- JTC1.22.19568: Library Fundamentals
 - Thomas Köppe (editor)
 - Jeffrey Yasskin (backup)

Complete projects:

- JTC1.22.14882:1998 and :2003, Programming Language C++:
 - Andrew Koenig (editor)
 - Tom Plum (backup)
- JTC1.22.14882:2011, Programming Language C++:
 - Pete Becker (editor)
 - Lawrence Crowl (backup)
 - Tom Plum (former backup, until 2006)
- JTC1.22.14882:2014, Programming Language C++:
 - Stefanus Du Toit (editor)

- Lawrence Crowl (backup)
- JTC1.22.14882:2017, Programming Language C++:
 - Richard Smith (editor)
 - Thomas Köppe (backup)
- JTC1.22.14882:2020, Programming Language C++:
 - Richard Smith (editor)
 - Thomas Köppe (backup)
- JTC1.22.18015, Technical Report on C++ Performance
 - Lois Goldthwaite (editor)
 - Detlef Vollmann (backup)
 - Martin O'Riordan (former editor, until 2003)
- JTC1.22.18822: File System Library
 - Beman Dawes (editor)
 - Stefanus Du Toit (backup)
- JTC1.22.19217: Concepts
 - Andrew Sutton (editor)
- JTC1.22.19570: Parallelism
 - Jared Hoberock (editor)
- JTC1.22.19768, Technical Report on C++ Library Extensions
 - Matt Austern (editor)
 - Pete Becker (backup)
- JTC1.22.21544: Modules
 - Gabriel Dos Reis (editor)
- JTC1.22.23619: Reflection
 - David Sankel (editor)
- JTC1.22.24733, Technical Report on Extensions to Support Decimal Floating Point Arithmetic
 - Robert Klarer (editor)
 - Pete Becker (backup)
- JTC1.22.29124 Programming Language C++ - Special Math Functions
 - Walter Brown (editor)
 - Pete Becker (backup)
- JTC1.22.19216: Networking
 - Jonathan Wakely (editor)
- JTC1.22.19571: Concurrency
 - Michael Wong (editor)

- JTC1.22.21425: Ranges
 - Casey Carter (editor)

Cancelled projects:

- JTC1.22.19569: Arrays
 - Lawrence Crowl (editor)
- JTC1.22.24737, Technical Report on C++ Library Extensions
 - Matt Austern (editor)
 - Pete Becker (backup)

4.3. ELECTRONIC DOCUMENT DISTRIBUTION

WG21 has conducted much of its detailed technical discussion using the email lists provided by the Standard C++ Foundation via isocpp.org.

WG21 uses a secure wiki maintained by Edison Design Group. This secure wiki is used for quick exchange of documents during and between meetings.

WG21 is now providing all the appropriate committee documents electronically, eliminating the need for paper mailings.

4.4. RECENT MEETINGS

See isocpp.org/std/meetings-and-participation/upcoming-meetings for a list of recent and future meetings.

Proposal to permit paragraph numbers in International Standards

Date: 2023-08-09

Thomas Köppe <tkoeppe@google.com>, ISO/IEC 14882 (C++) project editor

JeanHeyd Meneide <wg14@soasis.org>, ISO/IEC 9899 (C) project editor

Malcolm Cohen <malcolm@nag-j.co.jp>, ISO/IEC 1539-1 (Fortran) project editor

Audience: ISO/IEC JTC1 SC22 national bodies, ISO JDMT

Proposal

ISO currently forbids the use of paragraph numbers in final publications. We propose to allow paragraph numbers to be used by documents that concern programming languages (e.g. documents originating from SC22).

Rationale

Standards for programming languages are highly technical and complex. It is of utmost importance for both the ISO working group members and the general public to be able to communicate precisely and efficiently when referencing the standard wording:

- When members of the public learn the rules of the programming language (usually filtered through several layers of educational material), there is ultimately a need for an authoritative reference to the standard.
- When authors of proposals to evolve the standard make technical arguments, they need to refer to the current wording.
- When users or vendors discover a possible defect, they need to describe precisely the applicable standard wording in relation to the observed and expected behaviour.
- When changes to a working paper are approved by SC22, clear and precise editing instructions need to be given to the project editor to locate the correct part of the wording that is to be modified.

For example, both the C (ISO/IEC 9899) and C++ (ISO/IEC 14882) standards contain numerous subclauses that comprise many paragraphs, sometimes spanning multiple pages. It is not uncommon for multiple paragraphs to be parallel and similar. In any discussion of the technical details, being able to refer directly to a paragraph in question is of great help. It would cause significant additional, entirely avoidable mental overhead if we could only refer to subclause numbers, perhaps requiring manual counting of paragraphs.

History

The C and C++ standards have used paragraph numbers in all their previous publications, based on individual case-by-case permissions. C++ even had national a body comment on the C++14 CD to extend the numbering to bulleted lists (comment “US-1”), which was a greatly appreciated usability improvement. Fortran has used paragraph numbers since the 2010 publication.

Alternatives

Instead of numbering paragraphs, we could use a further level of subclause: subclauses do not need to have a title, and we could convert each paragraph into its own subclause. However, this is unappealing for several reasons:

- The Drafting Directives limit the nesting depth of subclauses to 5, which we already exhaust in C++.
- Printing the full subclause number would create a significant amount of visual noise.

- Subclauses cannot be used for the individual items of a list.
- As a minor argument, the paragraphs in the current subclauses are often “narratively cohesive” and do not stand on their own. Therefore, conceptually they are better modelled as paragraphs than as distinct subclauses.

Proposed wording

In the [ISO/IEC Directives, Part 2, 2021](#), add a subclause below [22.3, Numbering, subdivision and hanging paragraphs] as follows:

22.3 Numbering, subdivision and hanging paragraphs

22.3.1 Numbering

[...]

22.3.2 Subdivision

[...]

22.3.3 Paragraph numbering

When a document defines a programming language, it is permitted for paragraphs to be numbered. Paragraph numbering shall be applied or not applied uniformly to all paragraphs of the document. Paragraph numbers shall be continuous and start at 1 within the directly containing subclause.

22.3.14 Hanging paragraphs

[...]