

Document No: WG21 N4694

Date: 2017-08-10

Project: Programming Language C++ – Extensions for Ranges

References:

Reply to: Casey Carter <casey@carter.net>

ISO/IEC PDTS 21425, C++ Extensions for Ranges, National Body Comments

Attached is a complete set of National Body Comments submitted to JTC1 SC22 in response to the SC22 Ballot for ISO/IEC PDTS 21425, Committee Draft of the Technical Specification for C++ Extensions for Ranges.

Document numbers referenced in the ballot comments are WG21 documents unless otherwise stated.

Responses to NB Comments for ISO/IEC PDTS 21425

Date:2017-08-10

Document: SC22/WG21 N 4694

Project: 21425

MB/ NC ¹	Line number	Clause/ Subclause	Paragraph/ Figure/Table	Type of comment ²	Comments	Proposed change	Observations of the secretariat
GB 1 001				Ge	Consider all outstanding issues before the final TS is produced.		Accepted. All P1 and P2 issues and most P3 issues are resolved in N4684. Other issues were deemed non-blocking.
CA1 002				GE	We do not have technical comments on the document, however, we note that the format of the document is inconsistent with the ISO IEC Directives part 2, and with the ISO Online Browsing Platform, which makes the following clauses available publically, clauses 1, 2 and three.	Restructure the document such that Clause 1 is scope Clause 2 is Normative References Clause 3 is Terms and Definitions Place other material in clauses 4, 5, 6, etc.	Accepted.
CH-1 003				ge	Consider adoption of the Concepts TS to ISO/IEC 14882:2014, given its relevance to other TSes. Building a house of cards of possibly volatile TSes is a disservice to the community and risks relevance and adoption of this TS.	Consider adoption of the Concepts TS to ISO/IEC 14882:2014.	Rejected: There was no consensus to adopt this change.
CH-2 004				ge	Open a NWIP for the application of this TS to the library part of ISO/IEC 14882:2014. Without this work item, relevance and adoption of this TS are at risk.	Open a NWIP for the application of this TS to the library part of ISO/IEC 14882:2014.	Rejected: There was no consensus to adopt this change.
US 005		01.04	6	Ed	Headers are not provided in "directories" – instead, we introduce a common prefix for the new headers.	Reword without reference to "directories".	Accepted. The sentence in question in [intro.namespaces]/6 reads "New header names are prefixed with experimental/ranges/." in N4684.
US 006		02.01.1		Ed	The wording no longer matches the more	Entirely replace this wording with the proposed	Accepted.

1 **MB** = Member body / **NC** = National Committee (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial

MB/ NC ¹	Line number	Clause/ Subclause	Paragraph/ Figure/Table	Type of comment ²	Comments	Proposed change	Observations of the secretariat
					thoroughly reviewed form in the C++17 Working Draft.	wording for C++17.	
US 007		03.03.2.3	3	Te	What does it mean for two customization point objects to be “equal”? Are they required to provide an overload for <code>operator==</code> ? The note on the Semiregular concept suggests otherwise: “[<i>Note</i> : The Semiregular concept is satisfied by types that behave similarly to built-in types like <code>int</code> , except that they may not be comparable with <code>==</code> .— <i>end note</i>]”	Provide a more nuanced wording that two such objects shall have identical behavior when used as described in their customization contract. Clarify if, for example, two objects can hold logging information tracking which one is called, as long as the logging info does not interfere with contractual behavior.	Accept with modification: [customization.point.object]/3 now refers to [concepts.lib.general.equality] to make it clear that “equal” here means substitutable into equality-preserving expressions.
US 008		03.04.1.1	1	Ed	Macros are not entities c.f. C++ standard, ISO 14882:3p3		Accepted. N4684 [organization]/1 and [contents]/1 now refer to “entities and macros.”
US 009		03.04.3.3	2	Ed	It is not clear what “effects are undefined” means from a standards perspective.	Reword, such as “the program has undefined behaviour if ...”	Rejected: “the effects are undefined” is used in many places in the C++ standard, the committee believes its meaning is clear.
US 010		04.01	3	Ed	This sentence would be redundant with a proper header synopsis.	Strike sentence once synopsis is added.	Accepted.

1 MB = Member body / NC = National Committee (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 Type of comment: ge = general te = technical ed = editorial

MB/ NC ¹	Line number	Clause/ Subclause	Paragraph/ Figure/Table	Type of comment ²	Comments	Proposed change	Observations of the secretariat
US 011		04.02		Ed	There is no synopsis for header <experimental/ranges/concepts>	Provide the synopsis for <experimental/ranges/concepts>. This will include setting the precedent for how to declare a concept in a header synopsis.	Accepted.
GB 2 012		04.02.2		Te	Is 'const' part of the type? In 4.2.2 Concept Same [concepts.lib.corelang.same] we read: "Same<T, U>() is satisfied if and only if T and U denote the same type." is the intention that Same<const T, T>() is true or false?	Clarify the meaning - perhaps in terms of std::is_same or using the same wording ("T and U name the same type with the same cv-qualifications")	Accepted.
GB 3 013		06.02.1		Ed	Ordering of Iterator requirements p8 defines 'reachable' and p10 uses it; it might be better if p10 is moved before the current p9.	In 6.2.1 swap over p9 and p10. Additionally, 'valid' should be italicised in the new p9.	Accepted with modification: N4684 instead moved p9 *before* p8.
GB 4 014		06.02.4	02.3	Te	Unspecified evaluation order for concept WeaklyIncrementable — If i is incrementable, then &+i == &i. However the C++ standard makes no guarantees about the evaluation order of the expressions in ==. (This also affects 6.2.12 p3.1)	Replace the C++ syntax '==' with 'is equal to'	Accepted.
JP 015		06.09.2.2	p1	ed	ranges::begin(t) is missing in "requires" for Range.	Add "ranges::begin(t)," to "requires".	Accepted.
GB 5 016		07.01	p10	Ed	Missing 'shall' in 7.1p10 "Implementations provide a mechanism to resolve this ambiguity in favor of the overload that takes two ranges."	Add 'shall' after 'Implementations'	Accepted with modifications: this paragraph has been struck from N4684.
JP 017		07.03.12	p3	ed	Use of "distance" in this subclause only would be inconsistent.	Change "distance(first1,last1)" to "last1-first1" as in other subclauses.	Accepted.

1 MB = Member body / NC = National Committee (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 Type of comment: ge = general te = technical ed = editorial

Responses to NB Comments for ISO/IEC PDTS 21425

Date:2017-08-10

Document: SC22/WG21 N 4694

Project: 21425

MB/ NC ¹	Line number	Clause/ Subclause	Paragraph/ Figure/Table	Type of comment ²	Comments	Proposed change	Observations of the secretariat
JP 018		07.04.4	p5	ed	Lack of consideration about projection in Complexity.	Add a description about projection.	Accepted: N4684 [alg.transform]/5 reads “Complexity: Exactly <i>N</i> applications of <i>op</i> or <i>binary_op</i> and the corresponding projection(s).”
US 019		08		Ge	No range-based overloads for the algorithms in the <numeric> header. There is no obvious reason to omit this latter part of the C++14 algorithms collection.	Add range overloads in <experimental/ranges/numeric> for: accumulate, inner_product, partial_sum, adjacent_different, and iota.	Rejected: There was no consensus to adopt this change.
US 020		A.2		Ed	Library convention is to document deprecated features entirely in the deprecated annex, rather than in the normative section, with a reference from the deprecated annex.	Move the specification of the deprecated forms of mismatch, equal, is_permutation, swap_ranges, and transform algorithms directly into Annex A.	Accepted.

1 **MB** = Member body / **NC** = National Committee (enter the ISO 3166 two-letter country code, e.g. CN for China; comments from the ISO/CS editing unit are identified by **)

2 **Type of comment:** **ge** = general **te** = technical **ed** = editorial