A proposed plan for Contracts in C++

Timur Doumler (papers@timur.audio)
John Spicer (jhs@edg.com)

Document #: P2695R1
Date: 2023-02-09
Project: Programming Language C++
Audience: SG21, EWG

Abstract

This paper proposes a tentative plan for making progress on contract support in C++. If adopted, this plan could allow SG21, EWG, and CWG to finalise the design and wording for Contracts in time for inclusion of this feature in C++26.

1 Motivation

We believe that having contract support in C++ would be extremely useful for many C++ developers, and that ideally, we should ship it in C++26. Of course, getting Contracts done in time for C++26 is an ambitious goal. But we believe that it can be done, and propose a plan on how to get there. We further propose to take a poll in SG21 on officially adopting this plan. This would of course not be an irrevocable commitment: if things get delayed for whatever reason, Contracts will miss the train and we will have to wait three more years. But we believe that having a clear plan approved by SG21 will help to keep the group on track.

Progress on Contracts has been slow. After Contracts were removed from the C++20 draft, we needed to establish SG21, and agree on a path forward. Significant progress has been made as reflected in the Contracts working paper [P2521R2]. This involved the development of a set of use cases and eventually to the Contracts working paper, which resolved a number of the issues required to produce a minimal viable product (MVP) proposal.

Most recently, progress has been blocked since March 2022 because of controversy around the semantics of side effects in contract annotations. Several different approaches have been proposed (see [P2521R2] and references therein). Should side effects in contract annotations be allowed? If yes, should side effects be allowed to evaluate twice, or not at all? Or should they be undefined behaviour? Or should they make the program ill-formed? In case of the latter, how would the implementation detect this?

There are advocates for yet another possible approach where side effects are only allowed in pre- and post conditions if they stay within their own cone of evaluation. However, SG21 was unable to make progress on this issue because we needed to have a paper describing this approach before taking any action on it. A paper was produced for the October 2022 mailing (see [P2680R0]), but many details about the proposed approach are still left to be described.

The plan proposed in this paper is designed to prevent similar delays from happening in the future. According to this plan, the outstanding design decisions need to happen in SG21 by a certain date. If a contributor pledges to produce a design proposal, but that proposal has not been provided in the form of a sufficiently elaborated paper that SG21 can vote on by the agreed upon date, it will not be considered.

2 The plan

Assuming that the relevant deadlines in the C++26 release cycle will be the same as they were for C++20 and C++23, the plan for getting Contracts into C++26 can be derived more or less directly from those deadlines. This is the plan we propose:

Meeting	**C++26 milestone**	**By this meeting, SG21 should…**
2022.3 Kona		Decide whether we want to consider the P2680 approach to side effects in contract annotations
2023.1 Issaquah		Make design decision on how to handle side effects in contract annotations in the MVP
2023.2 Varna	C++26 WP opens	Make design decision on contract violation handling modes in the MVP
2023.3 Kona		Make design decision on syntax (attribute-like vs. lambda-like) in the MVP
2024.1		Resolve any other outstanding design questions in SG21; forward complete MVP design to EWG for review and to LEWG to evaluate library impact
2024.2		MVP design review in collaboration with EWG (and LEWG if necessary)
2024.3	Last meeting for EWG review of C++26 features	Get MVP approved by EWG and send to CWG for wording review
2025.1	Last meeting to send proposals to wording review	Finalise MVP wording review in CWG; merge Contracts into C++26 WP
2025.2	C++26 CD finalised	Resolve any outstanding design/wording issues
2025.3		Resolve NB comments
2026.1	C++26 DIS finalised	Resolve NB comments

For simplicity, this plan is organised using physical WG21 meetings as milestones. However, we expect that the majority of the work will actually happen in between those meetings. The progress of this work will be reviewed in teleconferences and on the reflector.

In SG21, we need to make decisions on three outstanding major design questions: side effects, syntax, and contract violation handling modes. Then, we need to finalise the design, and forward this to EWG, which needs to review and approve this design. Then, wording has to be produced and forwarded to CWG, which needs to review and approve it. After this, Contracts can be merged into the WP. We should aim to achieve these milestones not at the last possible meeting, but leave a buffer of at least one extra meeting in case anything goes wrong. This leads to the plan shown above.

In order to track and document the current state of the agreed-upon design throughout the different stages of this plan, we should use the Contracts working paper [P2521R2] and keep it up to date.

3 Standard library impact

The proposed plan assumes that the C++ standard library will, at least initially for the MVP, not have any contract annotations itself. Therefore, the roadmap currently does not contain any provisions for LEWG and LWG to go through the whole standard library and add such annotations. However, we will still need to evaluate whether anything needs to be added to the language support part of the standard library in order to enable the core language feature. Therefore, the roadmap does contain an item for LEWG to evaluate library impact.

Should the design direction of the Contracts MVP change significantly, and should SG21 decide that contract annotations on standard library facilities should be part of an MVP, then this roadmap will be amended accordingly.

Polls

SG21 meeting in Kona, 2022-11-11

Adopt D2695R0 as presented as the roadmap for working on a Contracts MVP targeting the C++26 IS, which will need consensus in SG21 to revise. Consensus on technical decisions takes precedence over the roadmap; we will amend the roadmap accordingly if a technical decision requires it.

```
SF| F| N| A|SA
10|10| 5| 1| 1
```

Result: **Consensus**

SG21 meeting in Issaquah, 2023-02-08

Swap the target dates for producing a design for syntax and violation handling.

```
SF| F| N| A|SA
11| 3| 5| 0| 0
```

Result: **Consensus**

Document history

R0 → R1

- Swapped the target dates for producing a design for syntax and violation handling, as directed by SG21 poll

References

[P2521R2] https://www.open-std.org/jtc1/sc22/wg21/docs/papers/2022/p2521r2.html
[P2680R0] https://www.open-std.org/jtc1/sc22/wg21/docs/papers/2022/p2680r0.pdf