

Document Number: P2659R0  
Date: 2022-10-14  
Revises:  
Reply to: Brian Bi  
Bloomberg  
bbi10@bloomberg.net

## A Proposal to Publish a Technical Specification for Contracts

We propose to publish a Technical Specification for Contracts based on these sources of wording:

- [P0542R5] and [P1323R2], which were incorporated into the C++20 Working Draft before being removed by [P1823R0], plus minor changes introduced by the project editor;
- [P1607R1], which was approved by EWG, and was subsequently in the process of being reviewed by CWG when Contracts were pulled from C++20; and
- [P1344R1], which was approved by CWG, but not yet applied at the time when Contracts were pulled from C++20.

Publishing a Contracts TS would pave the way for implementations to begin providing users with the ability to write contracts, which would in turn allow users to start gaining much wider real-world experience. We hope that such experience will more expeditiously result in consensus over what the Contracts feature should comprise so as to be most useful to a wider range of C++ programmers, which to date has been manifestly difficult to achieve.

We note that, as of now, SG21 has not produced a complete proposal, and is on track to produce only a minuscule MVP that itself has been stalled for going on a year now. In contrast, most of the features in the proposed Contracts TS already have considerable implementation experience (see [P1680R0]). We believe that publishing a TS would be of enormous value in helping to advance the state of Contracts so that we can land a useful viable product in C++26.

The initial draft of the proposed Contracts TS will follow in a separate paper, [P2660R0]. We have deliberately kept it as close as possible to the aforementioned sources, making changes only where strictly necessary to produce a coherent whole that is based on the C++23 DIS, in order to have a starting point that consists only of features that were already approved by EWG<sup>1</sup>. Another, separate paper, [P2661R0], will propose to incorporate evolution since P1607R1.

---

<sup>1</sup>One notable exception is that we have chosen to rename the `inform` contract behavior to `observe` so that now, each of the four contract behavior names is a verb that has the contract as its direct object.

## References

- [P0542R5] G. Dos Reis, J. D. Garcia, J. Lakos, A. Meredith, N. Myers, and B. Stroustrup, *Support for contract based programming in C++*  
<https://www.open-std.org/jtc1/sc22/wg21/docs/papers/2018/p0542r5.html>
- [P1323R2] Hubert S.K. Tong, *Contract postconditions and return type deduction*  
<https://www.open-std.org/jtc1/sc22/wg21/docs/papers/2019/p1323r2.html>
- [P1344R1] Nathan Myers, *Pre/Post vs. Expects/Ensures*  
<https://www.open-std.org/jtc1/sc22/wg21/docs/papers/2019/p1344r1.md>
- [P1607R1] Joshua Berne, Jeff Snyder, and Ryan McDougall, *Minimizing Contracts*  
<https://www.open-std.org/jtc1/sc22/wg21/docs/papers/2019/p1607r1.pdf>
- [P1680R0] Andrew Sutton and Jeff Chapman, *Implementing Contracts in GCC*  
<https://www.open-std.org/jtc1/sc22/wg21/docs/papers/2019/p1680r0.pdf>
- [P1823R0] Nicolai Josuttis, Ville Voutilainen, Roger Orr, Daveed Vandevorde, John Spicer, and Christopher Di Bella, *Remove Contracts from C++20*  
<https://www.open-std.org/jtc1/sc22/wg21/docs/papers/2019/p1823r0.pdf>
- [P2660R0] Brian Bi, *Proposed Contracts TS*  
<https://www.open-std.org/jtc1/sc22/wg21/docs/papers/2022/p2660r0.pdf>
- [P2661R0] Brian Bi, *Miscellaneous amendments to the Contracts TS*  
<https://www.open-std.org/jtc1/sc22/wg21/docs/papers/2022/p2661r0.pdf>