

Modules for Standard C++

Gabriel Dos Reis Richard Smith

2018-05-07

Abstract

This document presents a set of changes to apply to the C++20 Working Draft to bring "modules" to C++ in stages. Simultaneously, it proposes a set of changes to the Modules TS Working Draft to incorporate ideas from the Atom proposal.

1 Introduction

At the Spring 2018 meeting in Jacksonville, Florida, the authors of this document put forward a plan of record [1] for bringing "modules" to standard C++. This document is our attempt to fulfill that promise made to the C++ community. The document, in its current form (as in the pre-Rapperswil mailing) is not complete. It represents a snapshot of our work at a certain point in time. In particular, there are issues related to "name lookup", "semantic property reachability", and "semantic effect availability" to harmonize. However, it contains the general overall structure developed during the Modules TS [2] and necessary to express the "modules extension" (Modules TS as amended by the Atom proposal [3]).

The issues related to "name lookup" are discussed in two companion papers. "Reachability of semantic property" is a notion developed as part of the Modules TS, necessary to handle "legacy headers" that are not "modular" (e.g. typical examples are unstructured C-style system headers) or for projecting modular views over existing headers.

Under the Modules TS, a semantic property (such as a class being defined or a function parameter having a default argument) is exported by a module if it is a property of an attendant entity of an exported declaration in that module, that is, if a walk over a simplified form of the semantic graph of the program starting from that exported declaration would encounter a declaration of the entity with the property, and the exported declaration occurs later than the declaration of the semantic property. A semantic property is reachable at a point within the program if it occurs earlier within the same translation unit or is exported by an imported module (or, for a module implementation unit, if it was declared in the purview of the module interface unit).

"Availability of semantic effects" is a notion from the Atom proposal that determines when semantic effects from a module (such as a class being defined

or a function parameter having a default argument) are usable, and serves a similar purpose to "reachability of semantic properties" in the Modules TS, but is computed in a different way. Under the Atom proposal, a module interface unit or module partition exports a semantic effect if that semantic effect either occurs within the translation unit or is exported by a module that the translation unit re-exports. A semantic effect is available at a point within the program if it occurs earlier within the same translation unit or is exported by a module interface unit or module partition that was imported into the translation unit.

References

- [1] Gabriel Dos Reis and Richard Smith. Plan of Record for Making C++ Modules Available in C++ Standards. Technical Report P0983R0, ISO/IEC SC22/JTC1/WG21, 2018. <http://wg21.link/p0983r0>.
- [2] International Organization for Standardization. *Programming Languages — Extensions to C++ for Modules*, 2018.
- [3] Richard Smith. Another take on Modules. Technical Report P0947R1, ISO/IEC SC22/JTC1/WG21, 2018. <http://wg21.link/p0947r1>.

A Merge into the C++20 Working Draft

B Merge into the Modules TS Working Draft