

C++ Club Meeting Notes

Gleb Dolgich

2018-05-17

- ▶ [Microsoft blog post](#)
- ▶ [MSVC Conforms to the C++ Standard](#)
- ▶ [Reddit](#)

- ▶ [p1062R0](#)
- ▶ [Reddit](#)

The proposed programmatic 2D drawing library is not a good fit for C++.

Reddit: Really think that the macro story in Modules is doing more harm than good

Modules and macros: a reasonable compromise (?)

- ▶ Post
- ▶ Reddit: “Nope”

What modules actually provide

► [Reddit](#)

C++ Modules and Large-Scale Development - John Lakos, ACCU 2018

Video

conference.accu.org

C++ Modules & Large-Scale Development

John Lakos

Thursday, April 12, 2018
This version is for ACCU'18, Bristol, UK.

2

@ACCUconf

accu
2018
April 11 - 14

C++ Modules and Large-Scale Development - John Lakos, ACCU 2018

► Bloomberg doesn't allow free functions (?!)

conference.accu.org

2. Introduce the Notion of a module in C++ Business Requirements for Modules

1. Introduction and Purpose

Modules are considered to be a **critically needed language feature** by many C++ developers, but the reasons for the urgency vary considerably from one engineer to the next. Some are looking, primarily, to **reduce protracted build times** for template-laden header files (e.g., with **build artifacts**). Others want to use modules as a vehicle to **clean up impure vestiges of the language, such as macros**, that leak out into client code. Still others are looking to **"modernize" the way we view C++** rendering completely — even if it means forking the language. These are all very different motivations, and they may or may not be entirely compatible, but **if the agreed-upon implementation of modules does not take into account established code bases**, such as Bloomberg's, **they will surly fall far short of wide-spread adoption by industry.**

131

@ACCUconf

ACCU
2018
April 11-14

C++ Modules and Large-Scale Development - John Lakos, ACCU 2018

conference.accu.org

2. Introduce the Notion of a module in C++ Business Requirements for Modules

In order for any new module technology to have a plausibly successful path to adoption, its **integration must be (purely) additive, hierarchical, incremental, and interoperable, but not necessarily backward compatible** with traditional rendering (e.g., `.h/ .cpp` pairs). By **(purely) additive**, we mean that providing a module-style interface to existing **code does not require that code to be modified** (in any way whatsoever). By **hierarchical**, we mean that what we **add to an existing code base** to provide module interfaces depends on that code base (and never vice versa). By **incremental**, we mean that adding a module interface to one part of the code **base never implies adding it to some other, disparate part** of the code base. Finally, by **interoperable**, we mean that a **C++ construct consumed through both a module interface and a (conventional) header-file interface** is understood by the client's compiler to be the same construct **without violating the ODR**.

135

@ACCUconf

ACCU
2018
April 11-14

Must-watch C++ talks

- ▶ [Reddit thread](#)
- ▶ [Matt Godbolt: What Has My Compiler Done for Me Lately? Unbolting the Compiler's Lid \(CppCon 2017\)](#)

List

Boost.Text: What a C++ standard Unicode library might look like

- ▶ [GitHub](#)
- ▶ [Docs](#)
- ▶ [Intro](#)

C++ is one of a select few major languages with no built-in support for Unicode. If nothing else, this is an embarrassment that should be remedied.

- ▶ [Release notes](#)
- ▶ [GitHub](#) – C++11, MIT
- ▶ [Comparison to other libraries](#)

Transwarp is a header-only C++ library for task concurrency. It enables you to free your functors from explicit threads and transparently manage dependencies. Under the hood, a directed acyclic graph is built that allows for efficient traversal and type-safe dependencies. Use *transwarp* if you want to model your dependent operations in a graph of tasks and intend to invoke the graph more than once.

C++ Committee Pre-Rappersvil mailing

- ▶ Mailing
- ▶ Reddit thread

P0709: Zero-overhead deterministic exceptions

- ▶ P0709R0
- ▶ Reddit thread

BEING A
PROGRAMMER

IS EASY. IT'S LIKE

RIDING A BIKE

EXCEPT THE BIKE IS ON FIRE,

YOU ARE ON FIRE,

EVERYTHING IS ON FIRE

AND YOU ARE IN HELL.