History

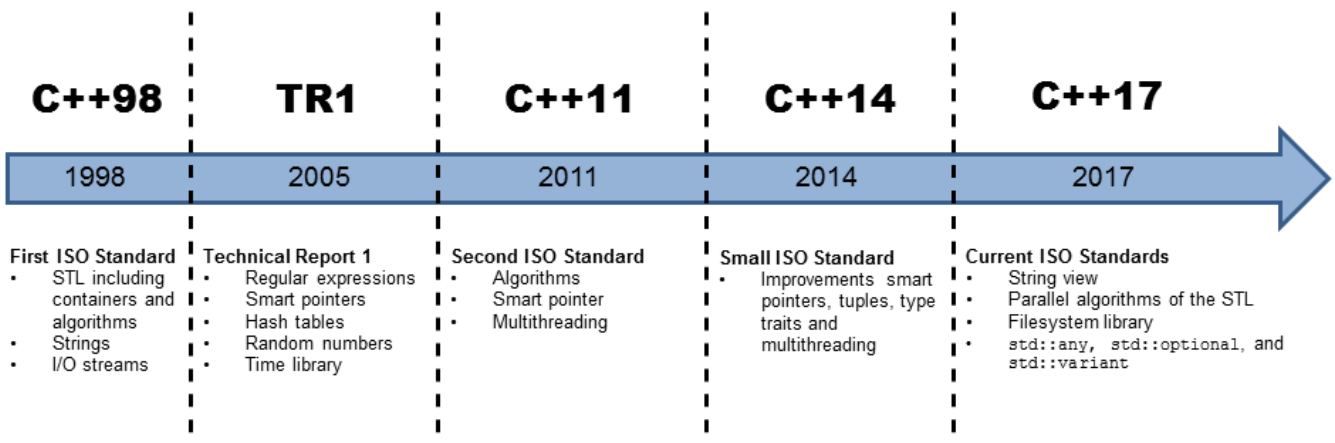
Let's dive into some of the milestones in C++'s long history.

WE'LL COVER THE FOLLOWING

The C++ standard library consists of many components. This chapter serves two purposes. It should give you a quick overview of the components and an idea of how to use them.

The History

C++ and therefore the standard library have a long history. C++ started in the 1980s of the last millennium and ended now in 2017. Anyone who knows about software development knows how fast our domain evolves. So 30 years is a very long period. You may not be so astonished that the first components of C++, like I/O streams, were designed with a different mindset than the modern Standard Template Library (STL). This evolution in the area of software development in the last 30 years, which you can observe in the C++ standard library, is also an evolution in the way software problems are solved. C++ started as an object-oriented language, then incorporated generic programming with the STL and now has adopted a lot of functional programming ideas.



The first C++ standard library from 1998 had three components. Those were the previously mentioned I/O streams, mainly for file handling, the string library, and the Standard Template Library. The Standard Template Library facilitates the transparent application of algorithms on containers.

The history continues in the year 2005 with Technical Report 1 (TR1). The extension to the C++ library ISO/IEC TR 19768 was not an official standard, but almost all of the components became part of C++11. These were, for example, the libraries for regular expressions, smart pointers, hash tables, random numbers and time, based on the boost libraries.

In addition to the standardization of TR1, C++11 got one new component: the multithreading library.

C++14 was only a small update to the C++11 standard. Therefore only a few improvements to the already existing libraries for smart pointers, tuples, type traits, and multithreading were added.

What comes next in the C++ standard library? With C++17 and C++20 we will get two new standards. C++17 is already done. C++17 includes libraries for the file system and the two new data types `std::any` and `std::optional`. With C++20 we might get libraries for network programming; with Concepts Lite we might get a type system for template parameters and better support for multithreading.

Now that we know the history of C++, let's talk about the various utilities it provides.