Best C++ Libraries to Fill the Gaps in the Standard Library

David January 9, 2015 Guides

C++ is sometimes criticized for its complexity and arcane syntax, but I think a bigger barrier to effective development in C++ is lack of access to high-level libraries, like the class libraries that come built-in with .NET or Java.

The C++ Standard Library is a great tool, including highly optimized algorithms and containers that are frequently needed when writing any application, but you don't see the networking libraries, XML libraries, crypto and many other facilities that you expect to have out-of-the-box with other high-level languages. Until C++11, there wasn't even a threading library, in spite of the importance of making good use of the multiple cores on modern computers if you want your application to scale.

This means that if you're writing a substantial application in C++, you'll probably want to be making use of some third-party libraries.

This article looks at the different functional areas you get with frameworks such as .NET and Java, and sees what libraries are available for C++ developers to serve those functions.

Page Contents

Library Collections

Before breaking down into the different library categories, it's worth calling out two libraries that span multiple functional areas:

- **Boost** – Boost is one of the best known and most used libraries in modern C++ development (after the standard library itself). It is, however, really a collection of many smaller libraries, and it doesn't aim to be the kind of coherent framework that you see in Java or .NET. You will likely want to pick and choose libraries from Boost for your project as some are more generally relevant than others.
- **POCO** – POCO isn't as well known as Boost, but is much closer in style and coherence to the class libraries for Java and .NET. It aims to provide high-level functionality in the same kind of key areas as other high-level language standard libraries do, whereas Boost libraries tend to be a bit more low-level, with a bit more work needed by the user to apply them effectively. That said, there is a tradeoff to be had between convenience in typical scenarios (leans in favour of POCO) and flexibility to do better optimization, handle special cases, and so on (which leans in favour of Boost).

Libraries by Category

User Interface

Desktop GUI

.NET equivalent: Windows Presentation Foundation (WPF), Windows Forms

Java equivalent: Swing, Abstract Windows Toolkit (AWT)

- **MFC**: The Microsoft Foundation Classes wrap up the Windows and COM APIs to allow for easier development of Windows desktop applications in C++. It is well supported in Visual Studio, which includes a visual forms editor. Royalty-free to use including for commercial use when your application is built with a suitably licensed copy of Visual Studio, subject to some restrictions. *Only for Windows of course!*
- **Qt**: The Qt application framework is a cross-platform library for GUI development, some general-purpose classes, and also an IDE (which includes a visual forms editor). Open-source and commercial licenses are available.
- **wxWidgets**: The wxWidgets GUI library isn't as polished as MFC or Qt, but if you are looking for a free cross-platform solution that you can use in commercial applications it may worth a look.

Web Applications

.NET equivalent: ASP.NET

Developing web applications in C++ is pretty far from the norm, but here are some options that are available:

- **CppCMS**: The CppCMS C++ web development framework aims to make web development in C++ convenient, and their site offers a rationale as to why you might want to develop web applications in C++.
- If what you're really after is a very basic embeddable HTTP server that you can adapt as a means to provide, for example, configuration and online help for a service or network-attached appliance, whose main purpose is not to be a general-purpose web application, then you will find various examples of simple C++ HTTP servers on the web. The cross-platform Boost Asynchronous I/O (ASIO) library contains examples of several different HTTP server implementations and the POCO C++ library contains a HTTPServer class.

Communications

Networking

.NET equivalent: System.Net namespace

Java equivalent: java.net package

- **Boost ASIO**: The Boost Asynchronous I/O library, which is also available in non-Boost form as a header-only library, is a cross platform library primarily for networking (TCP, UDP, ICMP and including IPv6 support), as well as some other I/O functionality including UNIX domain sockets and a wrapper around Windows API handles.
- **POCO::Net**: The POCO Networking classes provide a higher-level networking library compared to Boost ASIO, and in addition to the base networking protocols that Boost ASIO supports there are also classes for DNS, HTTP, FTP and email, and support for SSL/TLS.

Web Services

.NET equivalent: System.Web namespace

- **gSOAP**: The gSOAP toolkit is a mature but still-maintained library which supports SOAP and REST XML Web services.
- **Casablanca**: The Casablanca C++ REST SDK is a more recent (and less mature) creation by Microsoft to provide tools for accessing and authoring REST services to C++. In spite of the library's origin, it supports Linux, Mac OS X and several phone platforms. Given the rise in importance of REST for cloud services, this SDK may be one to watch.

Data

Databases

.NET equivalent: System.Data namespace

Java equivalent: java.sql package

- **POCO::Data**: The classes in the POCO Data namespace are a close match to the style of the database frameworks in .NET and Java, with a database-independent abstraction layer that wraps up communication with multiple databases. POCO::Data contains built-in support for MySQL, SQLite and ODBC.
- **OTL**: The Oracle, ODBC and DB2-CLI Template Library may be of interest if you are looking for an idiomatically C++ library, as this database abstraction is a header-only library with an STL-style interface.

XML

.NET equivalent: System.Xml namespace

Java equivalent: java.xml.* packages, org.w3c.dom, org.xml.sax

- **POCO::XML**: The POCO XML classes provide SAX2, DOM and an XML writer. Parsing uses the Expat XML C library.
- **Xerces C++**: The Xerces C++ library is a large XML framework. In addition to DOM and SAX2, it also supports XML Schema validation and with companion libraries built on top of Xerces, there is support for XML Stylesheet transformations and XPath 2/XQuery.
- You may also be interested in one of the fast, lightweight XML libraries for C++ of which there are several including TinyXML and RapidXML. These are not fully-featured XML libraries, but they are easy to use and integrate with.

JSON

.NET equivalent: System.Runtime.Serialization.Json namespace

- **JsonCPP**: The JsonCpp library is a simple JSON reader/writer for C++.
- **JSON Spirit**: The JSON Spirit library is another JSON reader/writer. It is built using the Boost Spirit parser generator, so you will need a dependency on Boost to use this.
- **Poco::JSON**: POCO has this one covered too with its JSON namespace.

Core Functionality

Multi-threading

.NET equivalent: System.Threading namespace, lock keyword

Java equivalent: java.lang.Thread, java.util.concurrent, synchronized keyword

- **Standard Library (C++11/C++14)**: Multi-threading is an area where the C++ standard was updated to include library support that was previously missing. Some basic functionality was added in C++11, and some more classes were added in C++14. If you have not yet adopted C++11, read on.
- **Boost Thread**: The Boost Thread library is an implementation (with different namespace) of the threading libraries now available in C++11 and C++14, so it makes that functionality available to those using older compilers.
- **POCO**: The POCO library has various classes to support multi-threading, including some higher-level abstractions for task management.

Filesystem

.NET equivalent: System.IO namespace

Java equivalent: java.io package

The C++ standard library has long had file I/O support, so this section addresses the *missing* filesystem functionality like directory iteration.

- **Boost Filesystem**: The Boost Filesystem library includes support for manipulating paths, checking for file and directory existence, copying and deleting, symlinks and directory iteration. This is likely to become part of the C++ standard in due course, and a compatible implementation may already be available in your compiler in the *std* namespace.
- **POCO**: The POCO library covers similar ground to the Boost Filesystem library in this area.

Cryptography

.NET equivalent: System.Security.Cryptography namespace

Java equivalent: java.security package

- **Crypto++**: The Crypto++ library provides the commonly used and some not-so-commonly-used cryptographic hash algorithms, symmetric block ciphers, asymmetric encryption and signing algorithms, pseudo random number generation and more. While nearly exhaustive in the algorithms it supports, the API style takes some getting used to.
- **Poco::Crypto**: The POCO Crypto namespace wraps up OpenSSL to provide easy-to-use classes for RSA, symmetric encryption, certificates and cryptographic hashes. POCO also contains a PRNG (pseudo random number generator) which wraps up the UNIX /dev/random PRNG or the Windows PRNG dependent on platform.
- Whilst these are C libraries rather than C++ libraries, you may also be interested in Microsoft CNG (Cryptography API: Next Generation) library which is a general purpose cryptography API if you are developing *for* Windows, or in the cross-platform OpenSSL library, which includes APIs for cryptographic algorithm primitives in addition to its well-known SSL/TLS functionality.

Regular Expressions

.NET equivalent: System.Text.RegularExpressions namespace

Java equivalent: java.util.regex package

- **Standard Library (C++11)**: Regular expressions are an area where recent versions of the C++ standard library have introduced support. If you aren't using C++11 yet, read on.
- **Boost Regex**: The Boost Regex library is a superset of the functionality that made it into C++11.
- **POCO**: POCO supports regular expressions along with other text formatting and tokenizing utilities.

C & C++.

← Using Python to Reduce JPEG and PNG Image File Sizes Without Loss of Quality

Assertions in C++, and Why Not to Use assert() →

Recent Articles

Enabling Crash Dumps

Addressing the XML/Object Impedance Mismatch – How to Generate Better Code from an XML Schema

Navigating the Visual C++ Runtime Library Variants

C++ API Design for SWIG

Making the Most of .NET XML Serialization

Search