What is the best open XML parser for C++? [duplicate]

Asked 16 years, 2 months ago Modified 1 year, 6 months ago

Viewed 371k times

255

This question already has answers here:

[What XML parser should I use in C++? [closed]](#) (6 answers)

Closed 8 years ago.

I am looking for a simple, clean, correct XML parser to use in my C++ project. Should I write my own?

c++ xml

Share

Improve this question

Follow

edited Jun 8, 2023 at 17:26

General Grievance

4,977 ● 37 ● 36 ● 54

asked Oct 4, 2008 at 17:21

whaledawg

2,615 ● 3 ● 16 ● 3

3 Note that the much newer StackOverflow posting I reference above has nearly as many upvotes as the current question (as of Dec 2014), and the answer has many more upvotes

than the answers here and has a fantastic, easy-to-read flow chart. – Dan Nissenbaum Dec 3, 2014 at 3:54

12 Answers

Sorted by: Highest score (default)

127

How about **RapidXML**? RapidXML is a very fast and small XML DOM parser written in C++. It is aimed primarily at embedded environments, computer games, or any other applications where available memory or CPU processing power comes at a premium. RapidXML is licensed under Boost Software License and its source code is freely available.

Features

- Parsing speed (including DOM tree building) approaching speed of strlen function executed on the same data.
- On a modern CPU (as of 2008) the parser throughput is about 1 billion characters per second. See Performance section in the Online Manual.
- Small memory footprint of the code and created DOM trees.
- A headers-only implementation, simplifying the integration process.
- Simple license that allows use for almost any purpose, both commercial and non-commercial, without any obligations.

- Supports UTF-8 and partially UTF-16, UTF-32 encodings.
- Portable source code with no dependencies other than a very small subset of C++ Standard Library.
- This subset is so small that it can be easily emulated manually if use of standard library is undesired.

Limitations

- The parser ignores DOCTYPE declarations.
- There is no support for XML namespaces.
- The parser does not check for character validity.
- The interface of the parser does not conform to DOM specification.
- The parser does not check for attribute uniqueness.

Source: wikipedia.org://Rapidxml

Depending on you use, you may use an XML Data Binding? **CodeSynthesis XSD** is an XML Data Binding compiler for C++ developed by Code Synthesis and dual-licensed under the GNU GPL and a proprietary license. Given an XML instance specification (XML Schema), it generates C++ classes that represent the given vocabulary as well as parsing and serialization code.

One of the unique features of CodeSynthesis XSD is its support for two different XML Schema to C++ mappings: in-memory C++/Tree and stream-oriented C++/Parser.

The C++/Tree mapping is a traditional mapping with a tree-like, in-memory data structure. C++/Parser is a new, SAX-like mapping which represents the information stored in XML instance documents as a hierarchy of vocabulary-specific parsing events. In comparison to C++/Tree, the C++/Parser mapping allows one to handle large XML documents that would not fit in memory, perform stream-oriented processing, or use an existing in-memory representation.

Source: [wikipedia.org://CodeSynthesis XSD]

Share Improve this answer Follow

edited Oct 4, 2008 at 21:30

answered Oct 4, 2008 at 19:49

jk.
6,468 ● 2 ● 28 ● 21

---

7 I like the headers-only approach (I think you really need one header file). Just throw it in and don't worry about changing anything in your build process. – Frank Feb 11, 2009 at 14:54

7 Hmmh. if "The parser does not check for character validity" and "The parser does not check for attribute uniqueness", it is, strictly speaking, NOT an xml parser -- these are not optional checks, mandated by xml spec itself. I would not waste my time on such a thing as there are actual good decent parsers too (libxml2 for example)_ – StaxMan Apr 23, 2009 at 4:06

3 It's the reason I use Rapidxml. One system I work with insists on putting illegal trailing spaces on the element names -

rapidXML is the only one that can cope with this (admittedly by not noticing!) – Martin Beckett Dec 1, 2009 at 1:50

1 rapidxml having many functionality to implement a xml ,like msxml .But node traversing is very difficult than other parser...and also file read and write ... – Rajakumar Jun 3, 2010 at 10:30

1 When choosing an XML parser for commercial use (in a certain kind of domain), we need to see if the parser will be maintained for at least 2 or 3 decades. Something like Xerces seems more likely to remain supported and maintained, than RapidXML. So would RapidXML be a wise choice to use? – Nav Jul 21, 2011 at 5:24

116

pugixml - Light-weight, simple and fast XML parser for C++ Very small (comparable to RapidXML), very fast (comparable to RapidXML), very easy to use (**better** than RapidXML).

Share Improve this answer Follow

edited Jul 7, 2016 at 3:31
svenevs
863 ● 9 ● 26

answered Aug 1, 2010 at 3:27
Zbyl
2,310 ● 1 ● 21 ● 27

29 Wow, that's a lot of claims. Can you back those up? What makes it better in those areas? Any reference articles? – Kissaki Sep 13, 2011 at 15:14

4 Reading a bit on the RapidXML as well as pugixml websites I understand what you (probably) mean. RapidXML is based on / inspired by pugixml. It has minimal documentation on

parsing. pugixml has good documentation on parsing and nice API. (Only read about parsing so far.) – Kissaki Sep 14, 2011 at 8:23

1 Pugixml is a lot easier to use, let's take reading xml from file - it's just load_file("file.xml")! I find it a lot more intuitive than rapid_xml. Selecting nodes by xpath also works pretty nice. – aurel Jun 18, 2012 at 6:58

I've been using pugixml for a few years. Works well, easy to integrate into projects, decent docs. BUT, no matter what package you use, XML composing/parsing in C++ is always a messy affair. – dlchambers Jan 14, 2013 at 21:28

2 @Kissaki I have tested a few XML parsers including a few commercial ones before using [pugixml] (pugixml.org) in a commercial product. – sg7 Jun 8, 2016 at 18:51

43

Try TinyXML.

http://sourceforge.net/projects/tinyxml

Share Improve this answer Follow

answered Oct 4, 2008 at 17:22

Rob
78.5k ● 57 ● 161 ● 199

2 Used tinyXML several times on VC++ and eVC++ - always worked fine – JohnIdol Oct 4, 2008 at 19:45

4 or use TinyXML 2 grinninglizard.com/tinyxml2/index.html – KindDragon Oct 17, 2012 at 16:30

I am trying this out, and for some reason the classes I call from tinyxml2 get a not resolved error. Any idea why? I found the classes in the header file which I included, so they should be available. – Evadecaptcha Dec 16, 2014 at 17:12

I have rejected this library (Also checked TinyXML2) from use because - library did not provide loading from unicode path names. Also currently I prefer usability and complete implementation over performance. – TarmoPikaro Sep 23, 2016 at 11:30

@KindDragon That link doesn't work anymore ("Account Suspended", d'oh!) – underscore_d May 30, 2023 at 14:05

16

TiCPP is a "more c++" version of TinyXML.

> 'TiCPP' is short for the official name TinyXML++. It is a completely new interface to TinyXML (http://www.grinninglizard.com/tinyxml/) that uses MANY of the C++ strengths. Templates, exceptions, and much better error handling. It is also fully documented in doxygen. It is really cool because this version let's you interface tiny the exact same way as before or you can choose to use the new 'ticpp' classes. All you need to do is define TIXML_USE_TICPP. It has been tested in VC 6.0, VC 7.0, VC 7.1, VC 8.0, MinGW gcc 3.4.5, and in Linux GNU gcc 3+

Share Improve this answer Follow

answered Oct 4, 2008 at 18:27

Kasprzol
4,149 ● 24 ● 20

The TiCPP link here says to instead use this one, and the `tinyxml` link doesn't work anymore ("Account Suspended", d'oh!) – underscore_d May 30, 2023 at 14:05

14

try this one: http://www.applied-mathematics.net/tools/xmlParser.html

it's easier and faster than RapidXML or PUGXML. TinyXML is the worst of the "simple parser".

Share Improve this answer Follow

answered Aug 18, 2010 at 9:55

Kat

141 ● 1 ● 2

1 They made a newer one: applied-mathematics.net/tools/IXMLParser.html – Andrew Jul 21, 2015 at 6:45

3 Just a warning though, to those who are checking it out as I am: the *newer* version has a really odd license and you can't even download it without first sending him an email. I think I'll go with pugixml. – Andrew Jul 21, 2015 at 6:51

12

Do not use TinyXML if you're concerned about efficiency/memory management (it tends to allocate *lots* of tiny blocks). My personal favourite is RapidXML.

Share Improve this answer Follow

answered Oct 4, 2008 at 19:48

yrp

4,575 ● 2 ● 26 ● 10

10

TinyXML can be best for simple XML work but if you need more features then try Xerces from the apache project. Go to the following page to read more about its features.

http://xerces.apache.org/xerces-c/

Share Improve this answer Follow

answered Oct 4, 2008 at 17:30

Raminder
1,887 ● 2 ● 18 ● 31

What features does Xerces have that TinyXML doesn't? – whaledawg Oct 4, 2008 at 17:36

OK, more to the point which of those features doesn't TinyXML have? – whaledawg Oct 4, 2008 at 18:06

It implements the whole DOM. TinyXML is simpler, but enough for keeping data in XML. – Lev Oct 4, 2008 at 18:48

4 Xerces implments the ENTIRe xml standard. TinyXML implments just enough to be useful. It turns out that 99% or users will only ever use 1% of the XML standard, so TinyXML is usually more that sufficient. – deft_code Jan 4, 2010 at 18:26

10

How about gSOAP? It is open source and freely available under the GPL license. Despite its name, the gSOAP toolkit is a generic XML data binding tool and allows you to bind your C and C++ data to XML automatically. There is no need to use an XML parser API, just let it read/write your data in XML format for you. If you really need a super-simple C++ XML parser then gSOAP may be an overkill. But for everything else it has worked well as testimonials show for many industrial applications since gSOAP was introduced in 2001.

Here is a brief list of features:

- Portable: Windows, Linux, Mac OS X, Unix, VxWorks, Symbian, Palm OS, WinCE, etc.
- Small footprint: 73KB code and less than 2K data to implement an XML web service client app (no DOM to limit memory usage).
- Fast: do not believe what other tools claim, the true speed should be measured **with** I/O. For gSOAP it is over 3000 roundtrip XML messages over TCP/IP. XML parsing overhead is negligible as it is a simple linear scan of the input/output while (de)serialization takes place.
- XML support: XML schema (XSD) import/export, WSDL import/export, XML namespaces, XML canonicalization, XML with attachments (MIME), optional use of DOM, many options to produce XML with indentation, use UTF8 strings, etc.

- XML validation: partial and full (option)
- WS support: WS-Security, WS-ReliableMessaging, WS-Addressing, WS-Policy, WS-SecurityPolicy, and other.
- Debugging: integrated memory management with leak detection, logging.
- API: no API to learn, only "soap" engine context initialization, then use the read/write interface for your data, and "soap" engine context destruction.

For example:

```
class Address
{
  std::string name;
  std::vector<LONG64> number;
  time_t date;
};
```

Then run "soapcpp2" on the `Address` class declaration above to generate the `soap_read_Address` and `soap_write_Address` XML reader and writer, for example:

```
Address *a = new Address();
a = ...;
soap ctx = soap_new();
soap_write_Address(ctx, a);
soap_end(ctx);
soap_free(ctx);`
```

This produces an XML representation of the `Address a` object. By annotating the header file declarations with

XML namespace details (not shown here), the tools also generate schemas. This is a simple example. The gSOAP tools can handle a very broad range of C and C++ data types, including pointer-based linked structures and even (cyclic) graphs (rather than just trees).

Hope this helps.

Share Improve this answer Follow

edited Jul 24, 2018 at 13:07
RAM
2,739 ● 3 ● 25 ● 54

answered May 12, 2010 at 19:23
Bob

3 For commercial use you have to pay one time fee for gSoap – Nayana Adassuriya Jun 6, 2013 at 5:33

9

TinyXML, and also Boost.PropertyTree. The latter does not fulfill all official requirements, but is very simple.

Share Improve this answer Follow

answered Oct 4, 2008 at 18:49
Lev
6,657 ● 6 ● 30 ● 29

2 `Boost.PropertyTree` was perfect for my kind of simple data storage. [This is the page]() that made it clear how to use it. Wow, I love boost. – Olical Nov 16, 2012 at 22:47

1 Boost PropertyTree is not that useful except in trivial XML files. The structure doesn't have backward linking so getting to parents of nodes means you really need to roll your own data structure to store the XML after Property Tree reads it. And it has no query support of the xpath nature. All you can do easily is read in an XML file into a tree structure and directly pull out a value if you know the exact path. – Minok Jun 3, 2014 at 20:45

I like the boost::property_tree too. There are some practical Visual Studio implementations of how to parse XML and JSON – AndyUK Jul 29, 2014 at 5:55

1 `boost::property_tree` is very bloated (increases compile time and executable size) and doesn't seem to be maintained anymore. Not recommended. – Andreas Haferburg Mar 25, 2015 at 10:34

I have rejected this library (Also checked TinyXML2) from use because - library did not provide loading from unicode path names. Also currently I prefer usability and complete implementation over performance. – TarmoPikaro Sep 23, 2016 at 11:30

9

I am a C++ **newbie** and after trying a couple different suggestions on this page I must say I like pugixml the most. It has easy to understand documentation and a high level API which was all I was looking for.

Share Improve this answer Follow

answered Sep 9, 2010 at 11:45

godspeedelbow

187 ● 2 ● 7

7

I like the Gnome xml parser. It's open source (MIT License, so you can use it in commercial products), fast and has DOM and SAX based interfaces.

http://xmlsoft.org/

Share Improve this answer Follow

answered Oct 4, 2008 at 17:59

dicroce
46.7k ● 31 ● 105 ● 149

You happen to be using CodeBlocks? Im trying to get the c++ wrapper for this up and running and it's giving me fits. – Dan May 22, 2020 at 18:51

2

Try TinyXML or IrrXML...Both are lightweight XML parsers (I'd suggest you to use TinyXML, anyway).

Share Improve this answer Follow

answered Oct 4, 2008 at 17:27

Prog
249 ● 1 ● 9